Biological Data Visualization: Analysis and Design

Ryo SAKAI

Examination committee:
Prof. dr. ir. H. Hens, chair
Prof. dr. ir. J. Aerts, supervisor
Prof. dr. ir. Y. Moreau
Prof. dr. ir. arch. A. Vande Moere
Prof. dr. ir. S. Aerts
Prof. dr. ir. J. Raes

Dr. ir. J. Reumers
(Jansen, Pharmaceutical Companies of Johnson and Johnson, Belgium)
Prof. dr. ir. J. Kennedy
(Edinburg Napier University, Scotland)

Dissertation presented in partial fulfilment of the requirements for the degree of Doctor in Engineering Science

May 2016

Preface

> "The greatest value of a picture is when it forces us to notice what we never expected to see." — John Tukey

Ever since the Information Visualization elective course in 2007 during my Master's in Biomedical Communications at the University of Toronto, I have been fascinated by data visualization. The creative and innovative works of Ben Fry, the enthusiastic and dynamic presentation of global development by Hans Rosling, and the classic visualization books by Edward Tufte were among my initial inspiration to pursue the study of data visualization. After two Master's degrees and one post-Master's degree, I had acquired visual communication skills, sufficient programming skills, and user-centred design principles to tackle the visualization research, but I struggled to find a researcher position with a particular emphasis on biological data. It was around August 2011 when I saw a posting for an open position in Prof. Jan Aert's group at KU Leuven. From that point onwards, everything went smoothly (except for some visa issues). It is hard to believe that more than four years have passed and that I am completing my Ph.D. thesis on data visualization of biological data.

The doctoral study initially focused on a project on visualizing structural variations of the human genome, but the study has evolved into a conglomerate of many design studies from a broad range of biological domains over years. This thesis examines the role of data visualization in data-intensive science. Each design study was revisited and reviewed to understand the intricate connection between the design process and the analysis. An extended framework for visualization and a practical guideline for visualization idiom design are presented. Many people have contributed either directly or indirectly to this study, and I am very grateful for all their support, contributions, and guidance.

First and foremost, I would like to thank Prof. Jan Aerts, who has given me the opportunity to pursue this visualization research and has taught me a tremendous amount by guiding research projects and introducing me to the

BioVis community. I realise that I may have been a bit stubborn at times, but I enjoyed our discussions on many projects. Prof. Aerts has also been very supportive of allowing me to attend international conferences, and I am grateful for the opportunities to be immersed in this field of research quickly. Discussions on future work in the evening after conferences have been some of the most fascinating and memorable moments.

I would like to extend my gratitude to my supervisors and examination committee members. Prof. Yves Moreau has provided critical and valuable comments at our regular YAC meetings, as well as during the preliminary defence. Prof. Andrew Vande Moere has always provided a novel designer's perspective when discussing design studies, and his comments have always challenged me in many ways to grow as a visualization researcher. Dr. Joke Reumers has been very supportive since our first visualization project (Pipit) and she has provided me valuable guidance in both science and technical aspects.

Furthermore, I have benefited greatly from the comments and suggestions of Prof. Stein Aerts and Prof Jeroen Raes for revising this thesis. I would also like to thank Prof. Jessie Kennedy, who provided valuable and extensive comments on an earlier draft of this thesis. Also, I am grateful to Prof. Hugo Hens for organising the preliminary defence discussion smoothly and punctually. I am privileged to have multidisciplinary experts on my examination committee, and I will value the experience of the very intense preliminary defence for the rest of my career.

Much of my work depended on collaboration with domain experts and many talented colleagues, who have taught me a great deal on the subject and technical methods. I would like to thank all BIOI members for a great workplace full of stimulating discussions and for all the fun we have had in the past four years. I enjoyed the lunch breaks, game nights, conference dinners, iMinds events, and learning each others' cultural differences. Special thanks to Jaak and Dusan for introducing me to various technical methods and discussing how we can visualize analysis outputs. I am also grateful to my current and former colleagues who worked together on various projects (Alejandro, Amin, Daniel, Georgios, Jansi, John, Nico, Peter, Raf, Thomas, Toni) and collaborators from the hospital (Ligia, Masoud, Niels, and Parveen). Further thanks to Alvin for all the insightful discussion on data visualization and the regular trips to Japanese restaurants in Brussels on weekends. I also wish to thank everyone at STADIUS, especially Elsy, John, Wim, Ida, Maarten and Liesbeth for the administrative and technical support.

During the doctoral study, I had opportunities to visit the Institute for Systems Biology (ISB) in Seattle, and the Broad Institute in Cambridge, Massachusetts. I would like to thank Dick Kreisberg for his support to realise the research visit

to ISB, and Vesteinn Thorsson, Sheila Reynold, and Prof. Ilya Shmulevich for the opportunity to join their team as a visiting researcher. I am also grateful to the Academische Stichting Leuven vzw for financial support for this visit. After conferences in Boston, I extended my stay to visit Bang Wong at the Broad Institute. I am most grateful for his hospitality and opportunities to work together on a number of visualization projects.

Last but not the least, my deepest appreciation goes to my family in Japan and the United Kingdom for their love and support.

Ryo Sakai

Bath, United Kingdom
May 2016

Abstract

Data visualization is an integral part of biological sciences and essential to enable dissemination of knowledge and sophisticated analysis of data. With advances in both biological data acquisition technologies and data-management and -processing technologies, researchers face challenges of developing better conjectures from the data that continue to increase in volume and complexity. Consequently, such data analysis often requires interdisciplinary expertise to address challenges in each case. In this thesis, we examine the design process of visualization projects from a wide range of application domains. The discussion includes the descriptive explanation of intermediate iterations towards the final design solution. The existing visualization model and framework are extended to characterise the design space of biological data visualization. Also, a practical 4-step guideline for visualization design is provided as an actionable evaluation method of visualization design. Careful retrospective analysis of each design case reveals that data visualization is ubiquitous, highlighting its vital role at different stages of data-intensive science.

Beknopte samenvatting

Data visualisatie is een integraal onderdeel van de biologische wetenschappen en essentieel voor een gedegen analyse van de gegevens en een verspreiding van de opgedane kennis. Mede door de vooruitgang in technologie voor zowel de generatie van data maar ook de verwerking ervan, worden onderzoekers steeds vaker geconfronteerd met grote uitdagingen wat betreft het formuleren van goede hypotheses en het testen er van. Een grote interdisciplinariteit is dan ook eigen aan het vakgebied en deze uitdagingen die zich stellen. In dit proefschrift onderzoeken we het design proces van verschillende visualisaties binnen een breed scala aan toepassingsgebieden. Dit onderzoek omvat de beschrijving van de verschillende iteraties als onderdeel van het design proces die leiden naar het uiteindelijke ontwerp. Het bestaande model voor het design proces wordt beschreven en uitgebreid. Verder wordt een praktische richtlijn beschreven in 4 stappen als basis voor een evaluatiemethode voor een visueel design. Uit ons onderzoek blijkt dat data visualisatie een vitale rol speelt in de moderne biologische wetenschappen en bij uitbreiding in vele takken van de wetenschap die te maken hebben met het verwerken van data.

Acronyms

AKL adenylate kinase lid.

BioVis Biological Data Visualization.

BOLD blood oxygenation level dependent.

CCRP Cosmopolitan Chicken Research Project.

D3 Data Driven Document.

DNA deoxyribonucleic acid.

EBI European Bioinformatics Institute.

eQTL expression quantitative trait loci.

EuroVis Eurographics Conference on Visualization.

GFF3 Generic Feature Format.

GO Gene Ontology.

GVF Genome Variation Format.

HCI Human-Computer Interaction.

IDE integrated development environment.

mRNA messenger RNA.

MSA multiple sequence alignment.

NCBI National Center for Biotechnology Information.

openGL open graphics library.

PCA Principal Component Analysis.

PDB Protein Data Bank.

PDF Portable Document Format.

RNA ribonucleic acid.

ROI region of interest.

SeDD Sequence Diversity Diagram.

SNP single nucleotide polymorphism.

SOM Self Organising Map.

SPLOM scatter plot matrix.

SVD Singular Value Decomposition.

SVG Scalable Vector Graphics.

UTR Untranslated Region.

webGL web graphics library.

Contents

Abstract v

Acronyms ix

Contents xiii

List of Figures xix

List of Tables xxv

1 Introduction 1

2 Framework and Model of Visualization 7

2.1 What-Why-How Framework 7

2.1.1 What 8

2.1.2 Why 8

2.1.3 How 10

2.2 Model of Visualization 14

2.3 Choice of Vis Tools 16

2.4 Custom Visualization Solutions 18

2.5 Card Sorting Technique 21

2.5.1 Abstract 21
2.5.2 Introduction 21
2.5.3 Related Work 23
2.5.4 Card Sorting 23
2.5.5 Case Study 26
2.5.6 Discussion 28
2.5.7 Acknowledgements 29

3 Visual Encoding Design **31**

3.1 Visual Analytics 31
3.2 Case Study: Fly Plot 34
3.3 Case Study: Pipit 40
3.4 Pipit: Visualizing Functional Impacts of Structural Variations . 49
3.4.1 Summary 49
3.4.2 Availability: 49
3.4.3 Introduction 49
3.4.4 Features 51
3.4.5 Discussion 52
3.4.6 Acknowledgement 52

4 Data Sketching **53**

4.1 Why Data Sketch? 53
4.2 Sequence Diversity Diagram - BioVis Redesign Challenge . . . 55
4.3 Sequence Diversity Diagram for Comparative Analysis of Multiple Sequence Alignments 65
4.3.1 Abstract 65
4.3.2 Background 66
4.3.3 Methods 69

4.3.4 Results 69
4.3.5 Conclusions 71

5 Sequential Tasks **73**

5.1 Case Study: Aracari 74
5.2 Case Study: Seagull 77
5.3 An eQTL Biological Data Visualization Challenge and Approaches from the Visualization Community 83
5.3.1 Abstract 83

6 Interaction Design **95**

6.1 Interaction 95
6.2 Case Study: Brain Constellation 98
6.3 Case Study: TrioVis 103
6.4 Case Study: Dendsort 105
6.5 TrioVis: a Visualization Approach for Filtering Genomic Variants of Parent-child Trios 112
6.5.1 Summary: 112
6.5.2 Availability: 112
6.5.3 Introduction 112
6.5.4 Features 114
6.5.5 Conclusion 115
6.6 dendsort: Modular Leaf Ordering Methods for Dendrogram Representations in R 116
6.6.1 Abstract 116
6.6.2 Introduction 116
6.6.3 Methods 118
6.6.4 Results 120
6.6.5 Discussion 127

6.6.6 Conclusions 128
6.6.7 Software Availability 128

7 Data Acquisition and Transformation 131

7.1 Introduction 131
7.2 Case Study: Oligoprobe 133
7.3 Case Study: Biplot Matrix 142

8 Beyond Desktop Applications 151

8.1 Introduction 151
8.2 Case Study: Fly Plot in Print 152
8.3 Case Study: CCRP 154

9 Conclusion 159

9.1 Conclusion 159
9.2 Lessons Learned 161
9.2.1 Skills and Knowledge 162
9.2.2 Design Study Guideline 162
9.2.3 Visualization as a Process 163
9.2.4 Environment and Work Culture 163
9.2.5 Design Contests 163
9.2.6 Practice in the Wild 164
9.2.7 Summary 164

Bibliography 165

List of Publications 181

9.3 As First Author 181
9.4 As Co-author 181

9.5 Awards 182

List of Figures

1.1 The four nested model for visualization design and validation . 3

2.1 Vis Design Space 10
2.2 4-step Vis Idiom Design Guideline 11
2.3 Ranking of Visual Variables based on Data Attribute Types . . 12
2.4 Example of Pattern Expressiveness 14
2.5 Model of Visualization 15
2.6 Vis Tools 17
2.7 Long-tail Distribution of Biological Research Questions 19
2.8 CellCyclePlot Interface 20
2.9 Card Sorting Result 28

3.1 Anscombe's Quartet 32
3.2 Reordered Anscombe's Quartet Data Tables 33
3.3 Fly Plot First Iteration 35
3.4 Fly Plot Second Iteration 36
3.5 Fly Plot Visual Encoding 37
3.6 Observed Patterns in Fly plot 38
3.7 Gene Modulation Dataset 39

3.8 Schematic Illustration of Structural Rearrangement Events . . . 41
3.9 Expert's Note . . . 42
3.10 Pipit First Prototype . . . 43
3.11 Pipit Second Prototype . . . 44
3.12 Pipit Third Prototype . . . 45
3.13 Pipit Third Prototype, zoomed in . . . 46
3.14 Pipit Visual Encodings . . . 47
3.15 Pipit Interface . . . 47
3.16 Pipit Collapsed View . . . 48
3.17 Pipit Expanded View . . . 48
3.18 Pipit Interface . . . 50

4.1 Sequence Logos of the Adenylate Kinase Lid Domain . . . 56
4.2 SeDD First Data Sketch . . . 58
4.3 SeDD Second Data Sketch . . . 58
4.4 SeDD Third Data Sketch . . . 59
4.5 SeDD Fourth Data Sketch . . . 59
4.6 SeDD Fifth Data Sketch . . . 60
4.7 SeDD Sixth Data Sketch . . . 60
4.8 SeDD Seventh Data Sketch . . . 61
4.9 SeDD Eighth Data Sketch . . . 62
4.10 SeDD Visual Encoding . . . 62
4.11 SeDD Design Process Overview . . . 63
4.12 Visualization Design Space . . . 64
4.13 Sequence Logo of the AKL Domain from Gram-negative Bacteria 67
4.14 Parallel Sets Representation of the AKL Domain . . . 68
4.15 Sequence Diversity Diagram of the AKL domain . . . 70

5.1 Aracari Gene Expression View 75
5.2 Aracari Visual Encodings for Distributions 76
5.3 Aracari SNP View 76
5.4 Sequence Diversity Diagram 78
5.5 Visualization of Mutual Information 78
5.6 Highlighting Selected Amino Acids in Jmol 79
5.7 Comparison of Mutual Information Vis Idioms 80
5.8 Different DNA-binding Specificity of Different MalR Transcription Factors 82
5.9 A Heatmap Representation of the Spiked-in Correlation Network in the Simulated Data 86
5.10 The Visualization Experts' Pick 91
5.11 The Biology Experts' Pick 92
5.12 The Overall Best Entry 93

6.1 Vis Design Framework: Interaction 96
6.2 Brain Constellation Data 99
6.3 Brain Constellation Data Transformation 100
6.4 ROI-wise Correlation 101
6.5 Brain Constellation Interface 102
6.6 TrioVis Interface 104
6.7 First Interactive Heatmap Prototype 106
6.8 Cluster Heatmap of Selected Pathways 107
6.9 Interactive Color Scale 107
6.10 Detail View Mode 108
6.11 Second Interactive Heatmap Prototype 110
6.12 Reordered Cluster Heatmap 110
6.13 European Cities and Comparison of Dendrogram Structures . . . 111

6.14 TrioVis Interface 113
6.15 Cluster Heatmap from TCGA 119
6.16 Hierarchical Clustering of a Simulated Two-dimensional Dataset 120
6.17 Recursive Algorithm for Ordering a Dendrogram Structure Based on the Minimum Distance 121
6.18 Comparison of Dendrograms from Different Linkage Algorithms Using R's Default Ordering Heuristics 122
6.19 Comparison of Dendrograms from Different Linkage Algorithms after Applying the MOLO Method Based on the Smallest Distance 122
6.20 Comparison of Leaf Ordering Methods in Cluster Heatmaps . . 124
6.21 Cluster Heatmap of the Data Matrix after Applying the MOLO Method Based on the Smallest Distance 127
6.22 Comparison of Dendrogram Structures Resulting from Different Leaf Ordering Methods 128
6.23 Cluster Heatmap of the Data Matrix after Applying the MOLO Method Based on the Average Distance 129
6.24 Comparison of Dendrogram Structures Resulting from Different Leaf Ordering Methods in a Limited Display Space 129

7.1 Vis Framework: Acquisition & Transformation 132
7.2 A Heatmap Visualization of a Transcript Expression Profile . . 133
7.3 First Prototype with Parallel Coordinates Plot 135
7.4 Second Prototype with Linked Views 136
7.5 Visualization of a SOM Output 136
7.6 Redesigned Visual Encoding of a SOM Output 137
7.7 Interface of an Integrative Vis Prototype 138
7.8 Alternative View: Heatmap 139
7.9 Alternative View: Small Multiples of Average Profiles 140
7.10 Search Function Based on the Functional Annotation of Genes . 141
7.11 Singular Value Decomposition of Simulated Gene Expressions . 143

7.12 Scatter Plot Matrix of Singular Vectors: 1 144
7.13 Scatter Plot Matrix of Singular Vectors: 2 145
7.14 Biplot Matrix: Drug and Compound Dataset 146
7.15 Biplot Matrix: Single Cell Dataset 147
7.16 Scaled Biplot Matrix 148

8.1 Fly Plot: Crowd Sourcing Exercise 153
8.2 Family Tree of Domesticated Chickens 154
8.3 Representation of Chromosome One 156
8.4 Chromosome Arrangement for Data Sculpture 156
8.5 Rendering of Data Sculptures 157
8.6 Installation of Data Sculptures 158

List of Tables

6.1 Comparison of the Total Line Lengths in Dendrograms 126

Chapter 1

Introduction

Data visualization is an integral part of biological science, and essential to enable dissemination of scientific knowledge and sophisticated analysis of data. In January 2007, Jim Gray presented his vision of the fourth paradigm of scientific research, describing how computing has fundamentally transformed the practice of science [1]. Gray explains that the scientific paradigm has evolved from experimental, theoretical, computational science to *data-intensive science*. As observed in today's molecular biology, the rapid advances in high-throughput experiments and high-performance data technologies attributes this paradigm shift.

With the advent of technologies to generate, collect, manage, and process data, the bottleneck in science is now our ability to analyse and derive hypotheses and insights from the data that continue to increase in volume and complexity. When an analyst has well-defined questions with respect to a dataset, they can employ computational methods, such as statistics and machine learning, to address their research questions. However, the accessibility and availability of more data than ever before tremendously increases the number of possible questions. Furthermore, analysis often does not start with well-defined questions in data-intensive science, because some questions are known while some are not. Thus, hypotheses are formulated via exploratory data analysis [2] to develop better conjecture aligned with a research interest.

Advanced computational methods help us find trends and patterns in large and complex data. However, much of their development and interpretation still requires a human in the loop. Also, the more complex the method gets, the fewer users who understand it. Hence, there is an opportunity for designing a visualization tool to support the development and to empower the user to

exploit the computational methods fully. In biology, for example, many of advances in data generation and collection result from automation, but the development of a computational approach and the interpretation still requires analysts [3]. Ultimately, the goal of biological data visualization is to support the development of analysis methods and the interpretation of results, encompassing both engineering and scientific research challenges.

Designing an effective visualization solution for a specialised domain is challenging because it requires the designer to acquire sufficient knowledge in the domain in order to understand the tasks of the user. According to research in cognitive psychology, we not only see passively registering information but also see actively on the demand of attention [4]. Hence, the user's previous knowledge and their intentions influence their analytical reasoning and visual thinking. In addition, a visualization designer needs to take account of terminologies and conventions that are unique to the domain, as well as semantics and metadata associated with datasets. The acquired domain-specific knowledge helps to analyse the data and tasks and ultimately informs the design of visualization systems. Biological data visualization is profoundly embedded in a specialised application domain, and previous knowledge on the subject is a prerequisite and assumed from the target user.

Evaluation of visualization systems is difficult because the appropriate choice of a metric depends on the task, and often tasks are ill-defined in exploratory data analysis. The four nested levels of visualization design model introduced by Munzner [5] defines the levels at which effectiveness of a visualization design can be evaluated (see Figure 1.1). The innermost level of algorithm relates to the computational side of visualization research while the outer levels are grounded in the design related subjects. For example, a visualization algorithm can be evaluated by analysing the computational complexity. On the other hand, the evaluation of design related attributes requires input from the users. In this thesis, each project stems from real-world problems, starting from the top domain situation level in the nested model. This approach is called the **problem-driven research**, and each case study is referred as a **design study**. A design study involves analysis of the domain problem, data, and tasks, implementation of a solution, evaluation of the solution with real users, and documentation of the findings [6].

Munzner provides an eloquent and succinct definition of visualization research:

> "Computer-based visualization systems provide visual representations of datasets designed to help people carry out tasks more effectively. Visualization is suitable when there is a need to augment human capabilities rather than replace people with computational decision-making methods. The design space of possible vis idioms

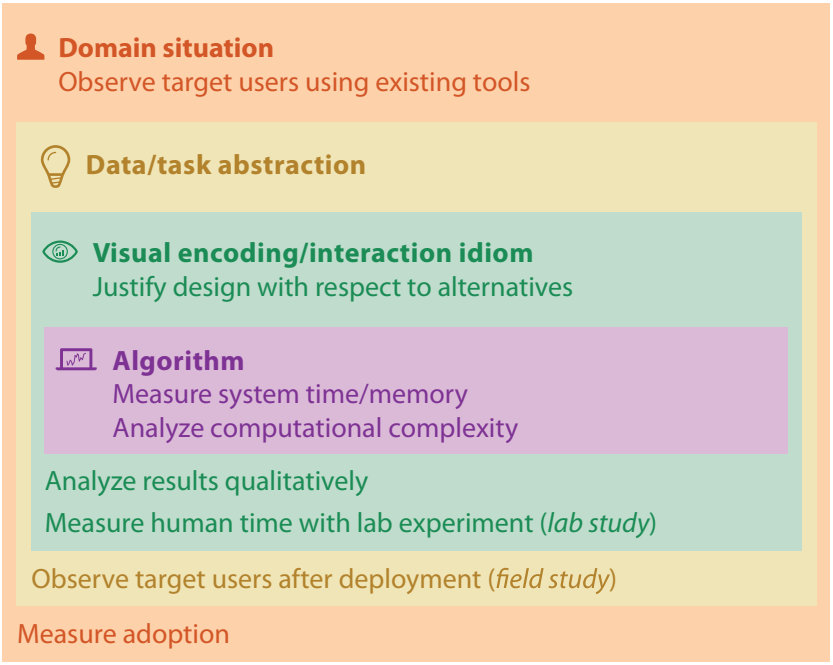

Figure 1.1: The four nested model for visualization design and validation. The original figure by Munzner [7] is licensed under CC-BY–4.0.

> is huge and includes the consideration of both how to create and how to interact with visual representations. Vis design is full of trade-offs, and most possibilities in the design space are ineffective for a particular task, so validating the effectiveness of a design is both necessary and difficult. Vis designers must take into account three very different kinds of resource limitations: those of computers, of humans, and of displays. Vis usage can be analyzed in terms of why the user needs it, what data is shown, and how the idiom is designed" [7].

As mentioned in the definition above, the visualization design needs to account for resource limitations of *computers*, of *humans* and of *displays*, which makes the study of visualization inherently multidisciplinary. It covers computer science, software engineering, perceptual and cognitive psychology and design-related research, such as graphic, interaction and user experience designs.

In the research presented, each design study involved development of a functional

visualization tool refined in subsequent iterations. I refer to these visualization tools as **prototypes**, instead of calling them "software" in order to differentiate their development process from the conventional software engineering process. Prototypes may be full of bugs and may not scale to larger datasets or a wider audience, but they are developed quickly to consider possible design ideas and to evaluate with the target user. The focus is similar to agile software development methods [8], and key concepts include *evolutionary development* and *early delivery*. Because the design space of visualization is vast and the majority of the possibilities are ineffective [7], the process of designing a visualization solution should be exploratory, quick and flexible to meet the analysis needs of the end user.

In the course of four years, there were collaborative projects with experts from a wide range of biological domains. They ranged from rare genetic disorders, single cell genomics, cancer genomics, neuroscience, proteomics and bioinformatics from the associated university hospital and other international research institutes. Some projects took unexpectedly longer than others. Collaborations started at varying stages in the overall project, some in early exploratory stages and some in later explanatory stages. Some projects led to publications, and some didn't. By carefully analysing projects from a wide range of application domains in retrospect, the existing theoretical frameworks and methodologies are extended to discuss caveats and design strategies for visualization projects involving biological data.

The thesis is a collection of design studies from a wide range of biological domains. The main research contributions are theoretical frameworks and practical methodologies for designing data visualization systems for biology. Instead of investigating an application domain in depth, the study examines various subjects in breadth to generalise strategies and solutions to visualization design. Thus, the key research questions for thesis are:

- What roles do data visualization have in data-intensive science?
- How do existing frameworks in visualization research extend to design studies from biological domains?
- What is a practical guideline for designing effective data visualizations for biological data?

This thesis is divided into chapters based on the common design themes which emerged from retrospective analysis. Within each chapter, each design study starts with a descriptive discussion of intermediate steps that led to the final design and is then followed by published papers. Because publications tend

to focus on the final design rather than the intermediate steps, the objective here is to elaborate on the transitional prototypes and the entire design process. The analysis structure is inspired by Munzner's book *Visualization Analysis & Design* [7], and the three-part What-Why-How analysis framework is used to examine design studies. The same terminologies and abbreviations are also used: visualization is written as **vis** for short, and a **vis idiom** refers to a distinct approach to creating and manipulating visual representation.

Chapter 2 starts with an introduction of the three-part What-Why-How analysis framework and an adapted version of the vis model by van Wijk [7, 9]. A practical guideline (**4-step vis idiom guideline**) is introduced to design a vis idiom by redesign. This chapter also provides an overview of types of data visualization and rationales for the custom data visualization tools to address the long tail questions. A brief summary of the landscape of existing programming languages and tools for data visualization is provided. We conclude this chapter with a card sorting technique paper to address the challenges of the domain characterisation.

Chapter 3 describes the visual encoding design process and the corresponding strategies. Visual encoding is central to data visualization design, and it involves mapping information to visual representations. Design studies on this topic include Pipit; a novel visual encoding for functional impacts of structural variations, and Fly plot; a feature level abstraction of dosage-specific drug response measured in gene expression levels.

Chapter 4 presents the concept of data sketching. The data sketching process involves rapid iterations to search and explore the vast design space of possible visual encodings. This process is illustrated with the design process of Sequence Diversity Diagram. Data sketching emphasises the early use of real data, early delivery of concepts, and iterative refinements.

Chapter 5 focuses on modes of analysis, where a sequence of tasks is the target of design. For example, Aracari, a visual analytics tool for expression quantitative trait loci (eQTL) data analysis, has two linked modes: gene expression view and SNP analysis view. Seagull, a visualization tool for comparative analysis of multiple sequence alignments of protein sequences, links three different views: Sequence Diversity Diagram, Circos visualization of mutual information, and 3D molecular structure viewer (jmol). By linking different views, it addresses a sequence of tasks that involves exploration as well as validation of insights.

Chapter 6 outlines the role of interaction in data analysis and data visualization design. The slogan of "Get it right with clicks" is introduced. Design studies in this chapter include Brain Constellation, a visual analytics tool for resting-state

functional MRI; TrioVis, a visualization tool for filtering genomic variants of patient-child trios; and dendsort, an R package for leaf ordering methods for dendrograms.

Chapter 7 describes how the computational and visual analytics approaches are combined to gain new insights into data. Computational or automated approaches, including statistical data mining and machine learning techniques, are essential to the analysis of big data. However, it inevitably creates a new challenge: with a more complex method, fewer users understand the method; thus resulting in unexploited potential of the method when introduced to new users. The challenges are addressed by the integration of computational methods and by providing a means to explore the output interactively.

Chapter 8 consists of visualization projects outside of desktop applications. In collaboration with scientists and artists, projects involved visualization design on paper and 3D data sculpture.

Chapter 9 consists of conclusions and discussions on future challenges and perspectives.

Chapter 2

Framework and Model of Visualization

Section 2.5.1 to 2.5.7 are reprinted from:

R. Sakai and J. Aerts, Card Sorting Techniques for Domain Characterization in Problem-driven Visualization Research, *Eurographics Conf. Vis.*, 2015.

Reprinted with permission from Eurographics.

2.1 What-Why-How Framework

When analysing a vis tool, it is useful to consider three high-level questions: (1) *what* is the input data, (2) *why* does the user need the vis tool, and (3) *how* does the vis idiom support the task? [7]. These three-fold **What-Why-How** questions correspond to the three key components of data visualization: data, tasks, and idioms. These components are interdependent of each other, and no vis tool is truly effective with one lesser component. For instance, no matter how well the vis tool is designed to support analysis tasks, if the data is of poor quality, the gain from the vis tool would be minimal. If a vis tool does not address the intended analysis tasks, the tool is pointless. When an ineffective vis idiom is chosen, the tool is not as effective as it could have been with a better vis idiom choice. Therefore, the evaluation of the problem domain using the *What-Why-How* framework is informative for the vis design process.

Interestingly, the writer and public speaker Simon Sinek introduced the concept of The Golden Circle [10], which also considers why, how and what questions. In *The Golden Circle*, *Why*, *How*, and *What* are arranged in a concentric circle starting with *Why* in the centre. Although the examples in the book are not from vis research, the general concept applies to vis research. Also, Sinek makes a compelling argument that we should first ask *why* for everything we do. Moreover, the breakdown of why-how-what questions is a simple and practical approach to design.

2.1.1 What

"What" is concerned with the input data that are used for visualization. One aspect to consider is the state of data, which may vary depending on the stage of the project. For example, a project may be in an early stage where only a sample dataset is available. Absent or a false promise of data is a common pitfall in vis design studies [6]. Or, a project may involve unprocessed experimental results, requiring a vis designer to learn how to process the raw data. Other projects may involve dynamic data instead of static data, where a vis tool needs to handle data streams. In a design study, it is not always possible to know how likely the data may change over the course of the collaboration, but it is useful to gauge the level of flexibility required for the development.

A methodological approach in visualization research is to abstract domain specific data attributes into domain-independent ones. This generalisation process is called **data abstraction**. There are five basic data types (items, attributes, links, positions, and grids), and there are three data attribute types (categorical, ordinal, and quantitative) [7]. Abstraction of data semantics and data types informs design choices, and it allows the designer to consider the use of existing vis idioms from other application domains.

2.1.2 Why

Why a vis tool is needed for a particular task is perhaps the most important question in a vis design study. If a task can be completed solely by computation, there is no need for vis [7]. However, the role of an analyst is irreplaceable for the interpretation of results and the subsequent decision making in research. The clear understanding of tasks is essential for a vis designer because it defines the goal and informs the design. A list of tasks also serves as a point of reference to evaluate how well a developed or existing system supports the intended tasks.

However, task analysis is not always straightforward because tasks are often not well-defined due to the exploratory nature of analysis, and understanding of tasks often requires domain specific knowledge, including conventions, terminologies, and semantics. The process of understanding the application domain is referred as **domain characterisation**, and it is a part of task analysis. Just asking the user to introspect about their analysis needs is largely insufficient [7], however there are a number of design activities that a vis designer can devise [11]. For example, **card sorting** is a participatory design exercise to achieve the shared understanding of the domain problem and analysis needs. The guideline and a case study of card sorting are described in detail in Section 2.5.1 to 2.5.7.

In biology, there are two main categories of tasks: **exploratory** and **explanatory**. This distinction is based on whether there is a message to communicate or not [12]. For example, a publication figure has a message that the author wishes to convey and communicate to a wide range of audiences. The goal of a figure is communication. On the other hand, in an exploratory data analysis, the researcher may know only partially or may not know what exactly the message is. Hence, the goal of exploratory visualization is analysis and hypothesis generation. Because of this difference in goals, the requirements, design considerations and strategies for vis design vary accordingly.

In the same way as *data abstraction*, domain specific tasks can be abstracted to domain-independent tasks. The process is called **task abstraction** [7, 13]. For example, the message in an explanatory figure boils down to a pattern or a relationship, be it a trend, an outlier, a cluster, correlation, or similarity. For exploratory visualization, the task can be generalised to identify, to compare, or to annotate patterns or relationships. Once you have identified a set of abstract tasks, you can prioritise the tasks, for instance using card sorting to define the hierarchy or order of tasks.

Another aspect of vis design space is whether the use of vis is *long-term* or *transitional* [7]. To prepare a figure for publication, one may start with a sketch and iterate on the figure to refine and improve communication of the message. To develop an interactive vis software, a developer may start with prototyping before committing significant time and effort to the software development. Figure 2.1 shows a conceptual design space of vis systems based on the nature of tasks and the scope of a vis tool. This thesis focuses on the lower left corner of this space, where vis tools are designed for exploratory analysis, and the use is largely transitional. These functional interactive vis tools are referred to as **prototypes** to distinguish from the traditional software engineering process. **Data sketches** refer to a design strategy of static visualizations with real data, discussed further in **Chapter 4**.

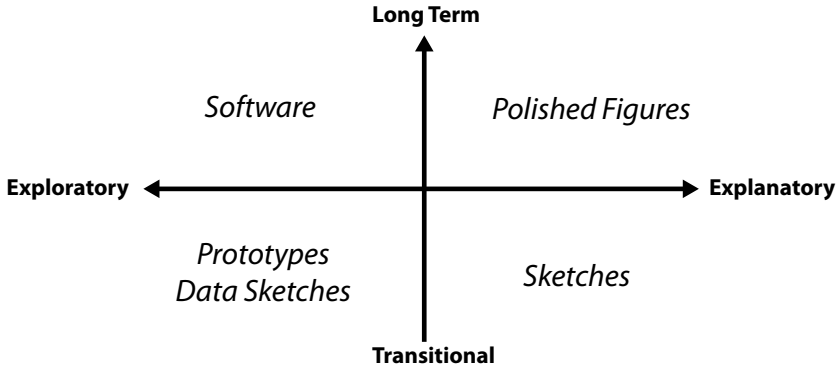

Figure 2.1: The vis design space on a spectrum of exploratory to explanatory tasks and a spectrum of long-term to transitional use.

2.1.3 How

A vis tool aims to support specific analysis tasks through a combination of visual encodings and interaction methods. "A distinct approach to creating and manipulating visual representation" is referred to as a **vis idiom** [7]. A vis idiom encompasses both static and interaction design choices. For example, the Sequence Diversity Diagram (SeDD) (discussed in Chapter 4) is a static visualization for multiple sequence alignments. This static vis idiom consists of a set of rules to translate sequence alignment data into a graphical representation. These rules are referred to as **visual encodings**. The SeDD was later extended to link other data and information via a user interface and interaction design (discussed in Chapter 6). A vis idiom describes the choice of visual encodings and interaction techniques.

The design process of a vis idiom is broken down into four steps (Figure 2.2), and this guideline is referred to as the **4-step vis idiom guideline**. The four steps consists of *Pop-out Effect*, *Effectiveness Principle*, *Pattern Expressiveness*, and *Interactive Exploration*. Each step has different design considerations, characteristics, and goals. This guideline is a result of reflecting on design studies in this research, and it is not intended to be a comprehensive guideline. Rather, it should be considered as a distilled and practical guideline that can be used to evaluate a vis idiom as a starting point. The basic principle of this guideline is "design by redesign", where a process starts with an evaluation of an existing or preliminary version. The division into four steps helps justify

design choices and evaluate possible options by weighing trade-offs.

	Pop-out Effect	***Effectiveness Principle***	***Pattern Expressiveness***	***Interactive Exploration***
Considerations	· Pre-attentive · Gestalt psychology	· Stevens' power law · Cleveland and McGill · Mackinlay	· Visual learning	· Interaction techniques
Domain Knowledge	Independent	Independent	Dependent	Dependent
Goal	Speed	Accuracy	Comprehension	Exploration

Figure 2.2: The 4-step vis idiom design guideline. The table shows example considerations, its dependency on the domain knowledge, the user's control, and the goal for each step.

The first step is called *Pop-out Effect*, which is concerned with the at-a-glance efficiency of visual processing. In 1988, the psychologist Ann Treisman systematically studied the properties of simple patterns that pop out from its surroundings [14]. This theoretical mechanism in a single eye fixation is called *pre-attentive processing*. Another consideration for the ease of search is a set of rules for pattern perception, called the *Gestalt laws*. The details of *pre-attentive processing* and the *Gestalt laws* are described in [15], and these principles help separate features (form, colour, motion, or spatial position) and examine visual distinctiveness. The evaluation of pop-out effect is independent of the user's domain knowledge, and our perceptual response is mostly involuntary. The goal of pop-out effect is to improve the speed and the ease of visual search.

For example, the sequence logos in Figure 4.1 use very distinctive primary colours to encode the functional groups of amino acids. However, the colour palette makes the figure visually overwhelming and consequently hinders reading the letters or comparing the height of letters, which is the main intended task for the figure. What is "easy" to see is what our perception is biased towards [4], and in this case, the pop-out effect does not improve the ease of visual search. Hence, the use of colour should be reconsidered in the redesign of the figure.

The second step is called *Effectiveness Principle*, which is concerned with matching the most important data attribute to the most effective visual channels. The measure of the effectiveness is the accuracy of our perceptual judgement against the objective measure [7]. This effectiveness criterion was introduced by MacKinlay [16] who was inspired by the systematic treatment of visual attributes by Bertin [17, 18]. This principle is grounded in the seminal research

in psychophysics [19, 20, 16], and more recently, Heer and Bostok extended the previous work by crowdsourcing graphical perception experiments [18]. As these empirical studies suggest, consideration of the impact of visual encodings on perceptual accuracy is a fundamental design strategy.

For example, the initial motivation for the Oligoprobe project (discussed in Chapter 7) stemmed from the *Effectiveness Principle*. The researcher used colour (hue) to encode quantitative values in a heat map. We saw an opportunity to introduce more effective and accurate visual encodings, such as the parallel coordinates plot, to improve visual analysis of the data. As Mackinlay's ranking of visual attributes based on data attributes [16] (Figure 2.3) suggests, the "Position" is more effective than the "Hue" to encode a quantitative value. Essentially, this diagram enables designers to optimise visual encoding based on the attribute types.

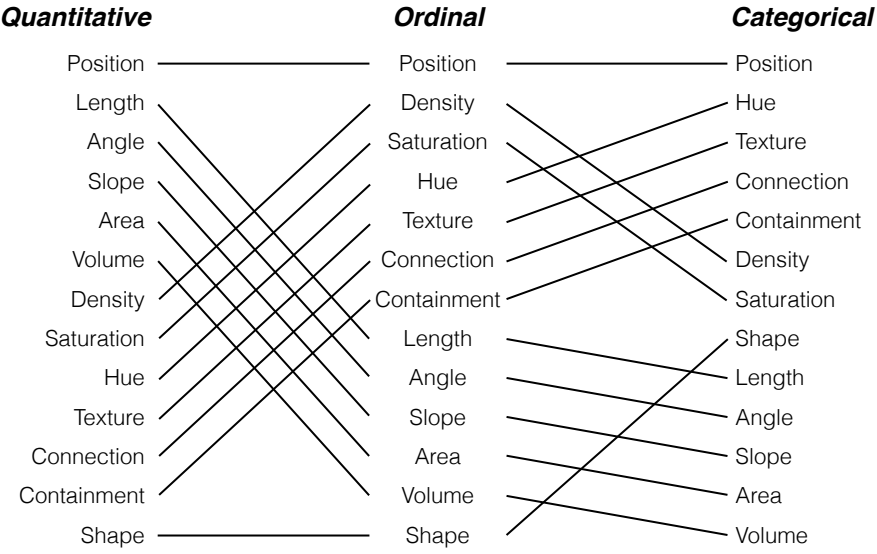

Figure 2.3: Ranking of visual variables based on data attribute types. This figure is based on the original figure of ranking of perceptual tasks by Mackinlay [16].

The third step is called *Pattern Expressiveness*, which is concerned with the interpretation of patterns. *Expressiveness* is another term introduced by Mackinlay along with *Effectiveness*, and it refers to the extent of how well a visual representation expresses the desired information [16]. One key concept in this step is *visual learning*, which largely depends on the user's previous experience and knowledge. The basic concept of *visual learning* is that we

can learn to interpret patterns better with practice [4]. An example of *visual learning* is the study by Fisher et al.[21], where they showed that analysts can learn to improve detection of statistically significant relationships in scatterplots with practice. In our design studies, we took the domain knowledge of users, including previous experiences and skills, into account for the design of vis tools. The goal of this step is to achieve a high level of *comprehension*, where visual encodings are designed to have patterns and relationships decoded effectively [22].

An example of *Pattern Expressiveness* in our design studies is the Fly plot vis idiom, discussed in Chapter 3. The Fly plot idiom conveyed the pattern of gene modulation scores at different drug dosages. The focus of the design was the overall pattern as seen in Figure 3.5, rather than encoding individual gene modulation scores as accurately as possible according to *Effectiveness Principle* as shown in Figure 3.3. The Fly plot is essentially a variation of the radial plot. Once the user learned the context and how to interpret the Fly plot, the users were able to identify both expected and unexpected patterns better in the design that focused on the *Pattern Expressiveness* (discussed in Chapter 8).

Another example of *Pattern Expressiveness* from literature is a novel visual encoding for the impact of drug class on a signaling network in different cell types (Figure 2.4), as discussed in [23]. The original figure was published in [24]. The figure leverages existing biological conceptual models to relate the organisational details of the experiment and allows the domain experts to assess the drugs' impact on the network. The figure uses the area and colour to encode quantitative values, which is not optimal according to the *Effectiveness Principle*. However, the figure is effective because it uses the spatial encoding to present the protein network and the design focus is on the patterns of multidimensional data presented in the biological context.

The last step is called *Interactive Exploration*, which is concerned with the design of user interaction to support the further exploration of the data. A user interaction triggers a change in representation. There are several frameworks and taxonomies of interaction techniques in visualization research literature [25, 26, 27, 28, 29]. In this thesis, we simply consider two key questions of interaction design: *how* does a user trigger the change and *what* are subsequent changes in the representation? The latter question is the main concern of interaction design in this thesis. The design of subsequent changes in the representation is guided by three previous steps of the *4-step vis idiom design guideline*. In Chapter 6, we introduce the slogan "Get it right with clicks" to limit the interaction to clicking in prototypes and examine the role of interactions via design studies.

The *4-step vis idiom design guideline* is not a set of rules that must be adhered to;

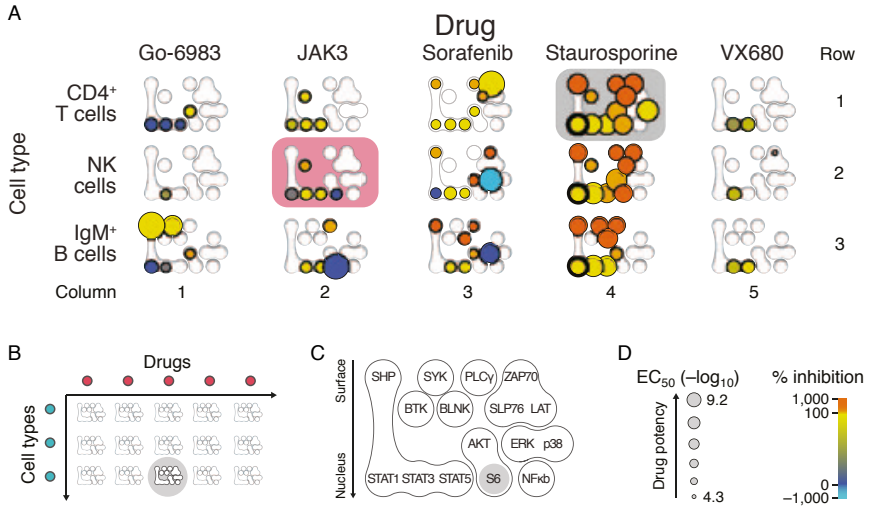

Figure 2.4: Overview of the impact of a drug class on a signaling network in different cell types. (A) Overview of the figure. (B) A layout of tabular small multiples representing experimental conditions. (C) The protein dimension is spatially encoded based on pathways. The vertical position relates to the intracellular position. (D) The coloured circles encode EC_{50} and percent inhibition using the hue. Adapted and reprinted from [23] with permission (License Number: 3740941414565).

rather it provides a set of questions that a vis designer can ask when designing a vis idiom. (1) *What are the elements that pop-out when you first look at the visualization?* (2) *Are there more perceptually accurate visual variables you can use?* (3) *Are the patterns easy to decode?* (4) *How does the interactivity augment our abilities in visual data analysis?* The first two questions relate to the perception and cognition, and the last two questions involve the user's knowledge and tasks. In the following section, we consider the interconnection among What, Why, and How components.

2.2 Model of Visualization

The **What-Why-How** questions can be further extended by combining it with the model of visualization by van Wijk [9]. Figure 2.5 depicts connections between relevant elements within the *What-Why-How* questions. As seen in van Wijk's model, the model consists of containers (white rectangles) and processes (grey rectangles) that transform inputs into outputs. One key addition to the

model is the **Acquisition & Transformation** process, which involves the acquisition of new data by means of conducting new experiments, integrating existing datasets, or transforming the data. This feedback loop to modify the input data is very common and imperative in biology. For example, a researcher may design a new experiment to validate their findings. Or, a researcher may use metadata to annotate functionality to interpret patterns or relationships they find. An example of data transformation is dimensionality reduction, where a high dimensional dataset is reduced to a lower-dimensional projection that retains most of its important structure [30]. Thus, our extended vis framework accounts for situations where the input datasets and the scope of analysis are changed iteratively.

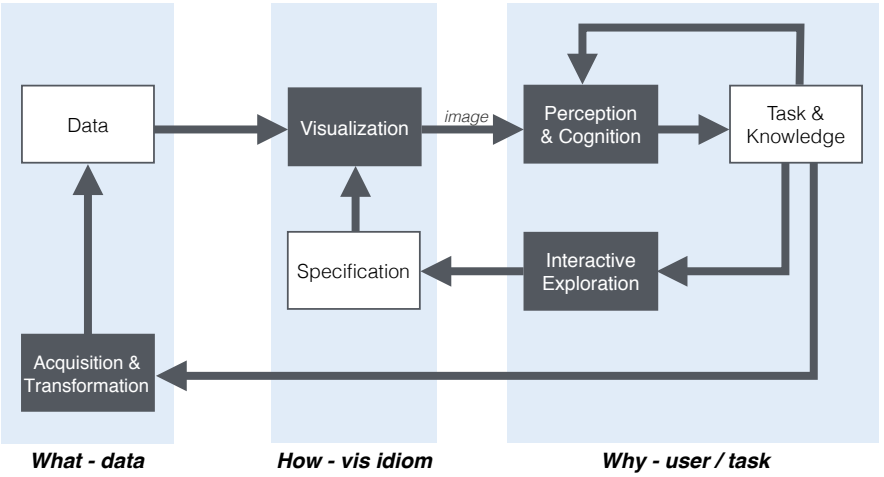

Figure 2.5: Model of visualization with respect to What-Why-How questions. This framework is an extended version of the model of visualization by van Wijk [9].

The rest of the containers and processes in the model is equivalent to the original van Wijk's model [9]. **Data** represents the datasets used for visualization. The **Specification** container includes parameter settings and algorithms for visual encodings and interactions. The **Visualization** process takes both **Data** and **Specification** as input and generates an image. The image is perceived and interpreted by the user, as represented by the **Perception & Cognition** process. The "Knowledge" container in the van Wijk's model is renamed as **Task & Knowledge** to emphasise the role of analysis tasks. **Task & Knowledge** influence how the user perceives and interprets the image (**Perception & Cognition**) and how the user interacts with the visualization (**Interactive Exploration**).

Task & Knowledge plays a critical role in biological data visualization. Depending on the tasks at hand and previous knowledge, the user may see differently due to reinforced relevant information in the top-down attentional processes [4]. For example, "P53, PTEN, BRCA1" may not mean much to a non-biologist, but they are abbreviations used for gene names, and more specifically they are well-studied tumour suppressor genes. Depending on the context, such as cell lines, experiments, and sample cohorts, these genes may have a subtle difference in meaning or nuance. Hence, the domain knowledge provides semantics to the data, which are often implicit to those who are already familiar with the domain. Also, with practice experts learn to interpret complex patterns rapidly and can identify patterns that non-experts fail to see [4]. As previously mentioned in the *4-step vis guideline*, this phenomenon is called *visual learning*, which is a part of **Task & Knowledge** in this model.

There are two feedback paths from **Task & Knowledge**. Based on the insights or conjecture gained from **Visualization**, the user may interact with the vis system to refine **Specification** to change how the **Data** is visualized. Or, the user may decide to acquire new data sets or apply computational methods to transform the input data. The role of **Interactive Exploration** is further discussed in Chapter 6, and the role of **Acquisition & Transformation** is further discussed in Chapter 7.

This model of visualization embodies the multi-disciplinary nature of visualization research, and each process and each container can be a focal point for vis research. For instance, the card sorting design exercise paper addresses the domain characterisation process involved in the **Task & Knowledge** element. Dendsort, an R package for leaf ordering methods, addresses dendrogram representations and their effects on matrix reordering as seen in the cluster heatmap technique, which corresponds to the **Specification** element. In the dendsort paper, we discuss how our methods influence **Perception & Cognition** as well as **Task & Knowledge**. The model described here is the basis for analysing each design study in the rest of this thesis.

2.3 Choice of Vis Tools

There are a number of tools to create visualizations. At the OpenVis Conference 2015, Jeff Heer presented a spectrum of vis tools with tradeoffs as shown in Figure 2.6 [31]. For example, Microsoft Excel can produce charts from tables of data fairly easily using their templates. While it is easy to use, the expressiveness of the output visualization is limited. On the other end of the spectrum, Processing is an open source programming language based on

Java and provides an integrated development environment [32]. The user of Processing will need to write code to visualise data, but the expressiveness of the vis outcome is not constrained by pre-defined templates. Each vis tool has its strengths and weaknesses.

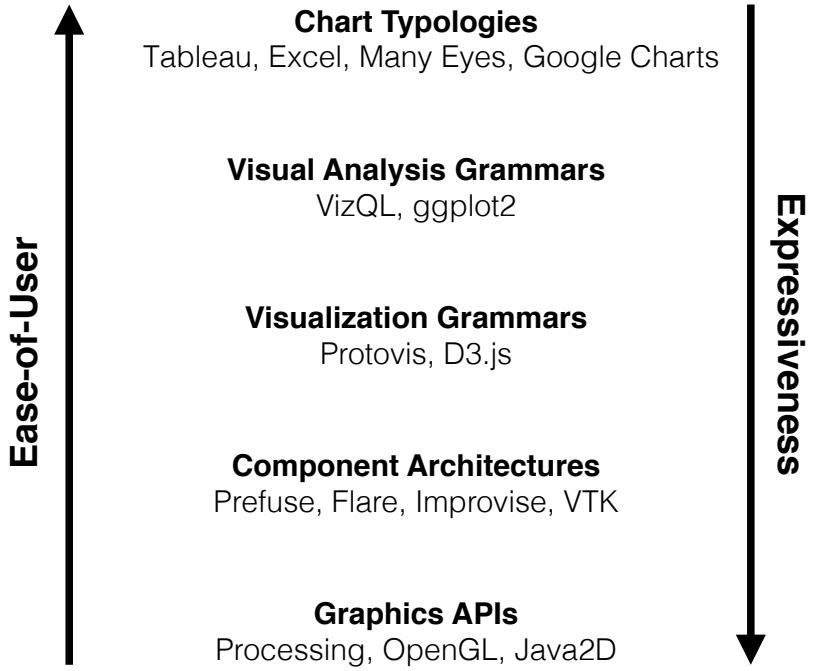

Figure 2.6: The spectrum of visualization tools based on ease-of-use and expressiveness. This figure is adapted from Heer's presentation at the OpenVis Conference 2015 [31].

The choice of a vis tool depends on a number of factors with respect to the task, the environment, and experience. For example, the deciding factors may include the developer's preference, whether the purpose is exploratory or explanatory, whether the desired output is static or interactive, whether the use is transitional or long-term, the target user's platform, and the sensitivity or security of the datasets.

In this study, most of the vis tools were developed in Processing [32] to explore the design space of possible visual idioms. Because the goal of a prototype is an early evaluation of the system, it is more important to be able to realise the idea quickly rather than making the tool accessible as web-based

applications. Processing also integrates the open graphics library (openGL) for accelerated 2D rendering, thus less engineering required to draw on a display than web applications where trade-offs of web graphics library (webGL), Scalable Vector Graphics (SVG), and canvas rendering need to be considered. Another advantage of Processing is the richness of existing Java libraries. For example, 3D data sculptures for the Cosmopolitan Chicken Research Project (CCRP) were generated in Processing using a library for 3D geometric objects (discussed in Chapter 8). The recent development of a javascript library (p5.js) that extends Processing to make coding accessible is also noteworthy [33].

In the Biological Data Visualization (BioVis) community, Data Driven Document (D3) is perhaps the most commonly used tool, especially with the general trend towards the development in web-based platforms. D3 is a JavaScript library that facilitates generation and manipulation of web documents with data [34]. Heer categorises D3 as a declarative language, where the programmer specifies what needs to be done instead of how to do it [31]. Although D3 is the state-of-art tool for software engineering to realise rich visual communications via the web, it was less fit for the work in this thesis because the focus was on the design process rather than software engineering.

2.4 Custom Visualization Solutions

Most collaborative projects in this study involved developing bespoke visualization tools to address unique and domain-specific research questions. Typically, a project involved one or two domain experts and meetings with them on a regular basis to refine vis design iteratively. The goal of these tools was to support their unique analysis needs as a result of new algorithms, experiments, or data integration methods. Although the uniqueness of their research questions limits the number of potential users, this uniqueness is what sets the researcher apart from another researcher working in the same domain. In a Data Stories podcast, Meyer refers to these questions as *long tail questions* [35].

The distribution of research questions in biology in terms of the domain specificity and the number of users is conceptualised in Figure 2.7. There are a larger number of users with general questions than with novel and special questions. Depending on the type of questions on this spectrum, the desired vis tool would be either a general purpose tool or a bespoke custom tool. Although both types of tools are indispensable in scientific research, the design approach and consideration for development are quite different depending on which type of vis tool is required.

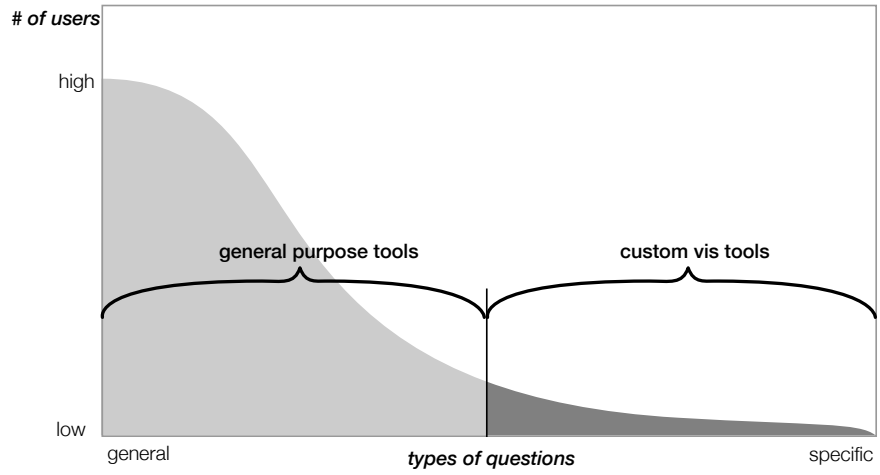

Figure 2.7: Long-tail distribution of biological research questions. For general questions, there will be more users and general purpose tools are more suitable to meet a wide range of common tasks. On the other hand, existing visualization tools may be insufficient to address the researcher's unique tasks, even though there may be only a small number of immediate users.

An example of custom visualization tools is CellCyclePlot (Figure 2.8). CellCyclePlot was developed for researchers who were developing novel algorithms for the copy number analysis in single cell sequencing. The tool took the output of their analysis workflow and provided an interactive visualization to aid genome-wide interpretation of the log R ratio and copy number data. It enabled the researchers to zoom and scroll through the data interactively. The extra information from the previous study, such as the known early duplication domain information, was also integrated. The key design consideration was to tailor for their data sets and their specific analysis needs.

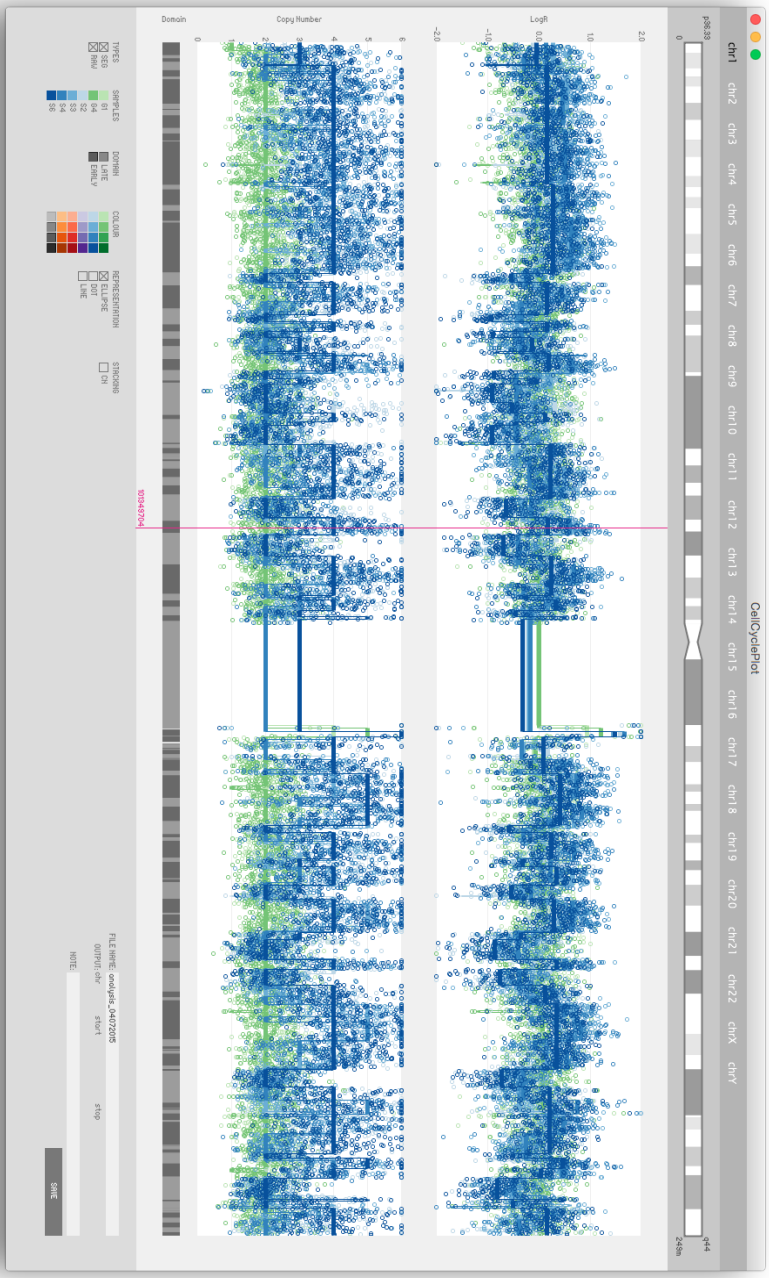

Figure 2.8: The interface of the CellCyclePlot application.

2.5 Card Sorting Technique

In every problem-driven visualization research, "Why" is the first question a vis designer need to address for understanding the tasks and the domain problem. This process is sometimes straightforward, but can be very complex due to the inherent nature of exploratory data analysis. In a collaborative project with computational biologists studying structural variations of the cancer genome, it was a struggle to prioritise and organise analysis tasks. This challenge motivated us for adopting the card sorting techniques from the Human-Computer Interaction (HCI) field and applying it to visualization research.

The following short paper (Section 2.5.1 to 2.5.7) was presented at Eurographics Conference on Visualization (EuroVis) 2015 in Cagriali, Italy. Reprinted with permission from Eurographics.

R. Sakai and J. Aerts, Card Sorting Techniques for Domain Characterization in Problem-driven Visualization Research, *Eurographics Conf. Vis.*, 2015.

2.5.1 Abstract

In problem-driven visualization research, the domain characterization is fundamental to the design process of a visualization solution to enable insight and discovery. Complex, fuzzy and exploratory analysis tasks in a specialized domain present considerable challenges to the designer, as well as the expert, to establish a shared understanding of the domain problem and analysis needs. In this paper, we provide a three-stage practical guideline for conducting a card sorting exercise to address challenges in the domain characterization and a case study from the biological domain.

2.5.2 Introduction

Establishing a shared understanding of the application domain and tasks presents considerable challenges for both a designer and a domain expert in problem-driven visualization research. The designer may struggle to build sufficient background knowledge in the domain to extract the expert's needs and to transform into more abstract low-level tasks. On the other hand, the expert may have difficulty articulating or introspecting about their needs because the domain tasks are complex and fuzzy due to the inherently exploratory nature of

the analysis and additional meta data available [7]. In addition, there may be other constraints, such as limited availability of the expert's time. We present a participatory design activity, namely card sorting techniques, to address challenges in the early stage of the design process.

Card sorting is a user centered design technique commonly used to elicit tacit grouping of items by asking respondents to sort a set of cards into meaningful groups [36, 37, 38, 39, 40]. For example, each card represents a component of a website, and these can be sorted by stakeholders to elicit categorizations as design implications and requirements for the website [37, 41]. Each card or item can be an object, a picture, or a name of attribute [42, 40], which are grouped in either *open* or *closed* sorting. In an *open* sort, the respondent names each resulting group themselves, whereas in a *closed* sort, a set of categories is predetermined and provided to the respondent. The choice of either *open* or *closed* sorting depends on the goal of the activity, whether to elicit tacit categorization of items, or to evaluate the assignment of items to categories. Thus, card sorting activity can be either *generative* or *evaluative* [11].

As the field of visualization matures with theories and models of the design process [36, 43, 5], we see a unique opportunity to narrow the focus to a specific stage in the process and provide practical guidance.We carefully analyze the existing guidelines and use cases of card sorting in literature from software engineering and human computer interaction [42, 44, 45, 46, 47, 40] and reflect on our experience to provide a practical and flexible guideline to address challenges in the domain characterization. We describe one case study where we collaborated with computational biologists to develop an interactive visualization system to study structural variation of the human genome.

In this paper, we focus on card sorting techniques, rather than the design study as a whole. Although techniques themselves are not novel, we highlight the flexibility and applicability of card sorting to a wide range of domains, and its usage as both generative and evaluative methods in the early stage of the visualization design process. By breaking down the card sorting exercise into 3 stages (*preparation*, *execution* and *analysis*), we describe options at each stage and provide practical advice.

In summary, the main contributions of this paper are:

- a three-stage practical guideline for conducting card sorting activities for the domain characterization
- a discussion of an exemplary case study from the biological domain

As other low-tech methods, such as "paper prototyping" or "wizard of Oz" [7], have been successfully adopted from the fields of software engineering and

human computer interaction by the visualization community, we anticipate that a wide range of readers from the visualization community would find the card sorting techniques useful and immediately applicable to address domain characterization challenges in their problem-driven projects, especially when the tasks are ill-defined and inherently exploratory. A card sorting activity helps to establish a shared understanding of the domain tasks and it takes us a step closer to reaching the "sweet spot" of gaining just enough domain knowledge and the tacit knowledge from the user to draw design implications and requirements [6].

2.5.3 Related Work

McKenna et al. [11] presents a design activity framework which consists of four overlapping key activities: "understand, ideate, make and deploy". This framework relates to the nested model [5] and provides actionable guidance throughout the visualization design process. Their paper also provides an extensive list of methods drawn from both the visualization community and the design literature. Card sorting is one of a hundred exemplary methods, and we elaborate on the application of this participatory design technique in the visualization design process.

Lloyd et al. [36] reports a successful use case of card sorting to categorize geovisualization domain tasks. Their exercise helped designers to gain an insight into varying spatial emphases in experts' approaches to tasks. Additionally, the comparison of sorting results between designers and experts allowed checking for the mutual understanding of the domain problem.

The special issue in Expert Systems (Volume 22, Issue 3, 2005) is a collection of papers describing the use of card sorting techniques and use cases in computer science. [42] gives a practical tutorial on sorting techniques. The collection also includes a wide range of analysis methods and case studies: a semantic analysis to investigate perception of women's office clothes [44], a method to derive co-occurrence matrices from card sorts to study perceived similarity of visual products [45], and statistical analysis techniques, such as the edit distance to measure similarity between two different sorts [46] and the orthogonality (aggregate difference) between two sorting results [47].

2.5.4 Card Sorting

The core activity of card sorting is to engage the participant to sort a set of items into categories [42, 40]. The original concept stems from the Personal Construct

Theory, which states that there is enough commonality to let us understand each other, but there are also enough differences to make us individual [48, 41]. Also, [49] points out that domain experts organize information based on abstraction of semantic characteristics, whereas novices organize information based on syntactic or non-domain specific characteristics.

In this paper, we target problem-driven visualization research, where "the goal is to work with real users to solve their real-world problem" [6]. Typically, this type of project involves a few domain experts from a specialized field and the number of accessible real users is often limited, at least at the beginning. Thus, we take a qualitative and a small scale approach, where each exercise is conducted on a one-to-one basis.

The same open card sort exercise can be repeated to gather a number of criteria and categories from a single respondent in one session. Also, you can recruit respondents with different roles, for example a "front-line analyst", a "gatekeeper", or a "tool builder" [6] to identify commonality or discrepancy in understanding of the domain problem. Depending on the design of the exercise, card sorting addresses different aspects of the domain problem.

In the following sections, we divide the process of card sorting activity into three stages (*preparation*, *execution*, and *analysis*) to discuss options and provide advice at each stage.

Preparation

The first task is to collect as much information as possible about the problem domain via conventional methods, such as contextual inquiry [50], observation and literature review. Based on your initial understanding of the domain tasks, you distill a series of questions that the user may ask in analysis and put each question onto a card. We call these entities, inquiry-based cards. In case of a complex question, consider decomposing into discrete questions. For example, a question may be, "When the value of A is higher than that of B, what is the value of C?" This question can be split into two separate questions: "Is the value of A higher than that of B?" and "What is the value of C?" Each question should be typed, printed, and stuck to an index card to improve legibility [42].

Besides analysis questions, the content of cards can be anything pertinent to the domain, including things that do or do not exist yet. For example, a set of cards may consist of data attributes and some which may have not been derived or acquired. As long as it is relevant and plausible, these items can be included and they may even encourage creative thinking.

Another type of card particularly useful for the visualization research, is a set of picture cards. The picture cards may consist of figures from the relevant literature, images from existing tools, and new visual encoding ideas. By introducing new visual encoding ideas, you can evaluate if the encoding is intuitive to understand or if it is appropriate in the context. Also, consider making each representation abstract enough that the user would understand the encoding, but would not be distracted by the details of the image [42].

The last advice is to look for design studies that characterize the same or a similar problem domain. Such design study papers may include the domain characterization as one of its main contributions.

Execution

Before conducting the exercise with a domain expert, the facilitator should carry out a session by themselves for two reasons. First, it allows the facilitator to familiarize themselves with items and to check if the collection of items is comprehensive to their knowledge. Second, the exercise will result in criteria and categories, which you can anticipate from the expert. The resulting categories can be the input for a closed sorting to identify commonality or difference.

Start a session by explaining about card sorting to the respondent. [42] provides sample instructions. Then, ask if the respondent understands each item and whether the set reflects domain tasks well. If some cards are deemed irrelevant to the task, those cards may be removed to be discussed afterwards.

We recommend a semi-structured format, where the respondent is guided, but allowed to deviate from the plan if necessary. For example, if the respondent remembers a relevant item in the middle of the exercise, allow them to add a new card to the set.

Once the respondent has grouped cards into groups, ask them to name each group and the overall criterion used for sorting. Then, discuss each group and criterion for clarification. Record the arrangement of cards by taking a picture with a smart-phone or a digital camera. We found it useful to index each item by numbering on the back, so that each card can be flipped in position and the number label is still legible in the photo. The same exercise can be repeated to elicit more groups and criteria.

Analysis

Given the scope of domain characterization and small-scale card sorting, we suggest semantic methods where interpretation of respondents' behavior and outputs relies on the facilitator's judgment [47]. A careful observation during the sorting exercise and analysis of resulting criteria and categories are critical to this approach. For instance, a respondent may struggle to sort an item. This item should be further investigated to understand the underlying source of difficulty. The semantic methods can provide rich insights, but requires the facilitator's time and scrutiny [47].

When analyzing criteria, categories and card assignment, it helps to compare commonality and difference between results from different respondents. Generally speaking, a high commonality suggests consistency, validity and usefulness of the categorization, while differences in categorization suggest inconsistency and variability. This process is instrumental to identify any potential discrepancy in understanding of the domain problem.

If there are a number of criteria obtained, each criterion can be further classified as either "subjective" or "objective" [42]. For example, "ones I like" is a subjective criterion, while "underlying data type" is an objective criterion. The card assignment is usually more consistent using an objective criterion than a subjective one.

2.5.5 Case Study

The case study is a collaborative project with computational biologists to develop an interactive visualization system to analyze structural variations of the human genome. The input data was preprocessed data from the whole genome sequencing of uterine cancer patients. Even after several interviews with the domain experts (the user), we (the designer) struggled to abstract the domain-specific tasks into system requirements.

The domain problem was fuzzy because the analysis tasks were complex, open-ended and inherently exploratory. In addition, a conventional ethnographic observation was not feasible, because the user worked on multiple projects concurrently. Having been unable to characterize the domain problem fully, we decided to adapt card sorting techniques to actively engage the experts in the participatory design exercise of one-hour sessions.

For the sorting exercise, we prepared two sets of cards: inquiry-based and pictures based cards. Based on the information we gathered from previous interviews and studying relevant literature [51, 52], we generated ten inquiry-

based cards and eight picture cards, three of which were our new visual encoding ideas. The questions and the collected figures are listed in the Supplementary Material 1.

Using these cards, we first practiced sorting exercise on our own, which resulted in the criterion ("genomic size and resolution") with four categories ("genome", "chromosome", "segmentation" and "feature"). These categories were consistent with the domain characterization from a design study of a multi-scale synteny browser [53]. We also prepared to use these categories for a closed card sort exercise to validate a shared understanding of the domain.

Although we scheduled with two "front-line analysts", only one respondent was available to participate in the activity. First, we asked the participant to examine the inquiry-based cards and see if the set represented questions asked during the analysis. At this point, the respondent added two questions to the set. Then, we asked them to conduct open card sorting with this inquiry-based card set.

This exercise took about 20 minutes to complete, and the respondent named the categories as "primary analysis", "in depth analysis", and "impact / validation" and its criterion as "process". Through the interview following sorting, it was confirmed that these categories reflected implicit stages in the analysis process. Neither the designer nor the respondent was aware of these stages, as also evident in the respondent's comment, "I have never thought of research questions this way, but it is interesting."

After recording the result of the open card sorting, we asked the respondent to perform closed card sorting. First, we used the same inquiry-based cards for closed sorting with the four categories based on the "genomic size and resolution". This took less than 10 minutes to sort, perhaps because the respondent was already familiar with the questions. Besides the two new questions added, the assignment of cards into provided categories was comparable to our sorting result. In the expert's result, some questions were placed between two categories, indicating that those questions could belong to either category.

Second, we asked the respondent to perform closed sorting with picture cards. The arrangement of inquiry-based cards from the preceding step remained on the table to link between questions and visual encodings. The result of this sorting session is shown in the Figure 2.9. Each card was flipped in position to show its unique label in order to record the arrangement (Supplementary Material 2).

There were two main outcomes of this card sorting activity. First, the categories of research questions based on the different stages in analysis informed us about the hierarchy and the order in which these questions were addressed in their

Figure 2.9: Result of closed card sorting with inquiry-based and picture cards. Each category is shown on a sticky note.

exploratory analysis. Subsequently, we reflected this order in the interface design of a prototype (Supplementary Material 3). The experts found the prototype useful and intuitive to use, and based on the insights gained, they advanced on to other research question before the prototype was fully developed into a software tool.

Second, careful analysis of the picture sort outcome indicated a gap in existing visual representations of structural variations. Because there was no picture card assigned to address the functional impact of structural variation at the feature level, this finding subsequently encouraged [54] the development of a novel gene-centric encoding of structural variation.

2.5.6 Discussion

Studying design processes in visualization design [36, 43, 5] and design study pitfalls [6] and reflecting on our experience, it becomes evident that establishing a solid understanding of the domain tasks and the problem is critical in the early stage of the visualization design process. We advocate the use of card sorting techniques because of their simplicity and adaptability to many specialized domains. The card sorting exercise actively engages the user in the design

process, and this type of participatory design exercise has also been shown to help establishing a rapport between the designer and the user [6, 55].

The advantage of card sorting is not only to elicit tacit categories, but also the distilling process in the card preparation. As complex tasks are decomposed into discrete items, it helps the designer understand the context as well as the relationships of tasks.

We found the use of inquiry-based cards and picture cards useful in our project, but these are not the only choice of entities or ways to run the exercise. In fact, the strength of the technique is that it is very adaptable to different purposes. In this paper, we do not discuss large scale card sorting and different analysis methods [38, 39, 49]. Because card sorting techniques are very versatile, we encourage other visualization researchers to share examples and anecdotal evidence of card sorting on the following web forum (`http://goo.gl/IPFsXu`).

2.5.7 Acknowledgements

This work was supported by the KU Leuven Research Council CoE PFV/10/016 SymBioSys, iMinds Medical IT ICON bSLIM, MyHealthData & MECOVI, and IWT O&O ExaScience Life Pharma.

The definitive version is available at (`http://diglib.eg.org/`).

Chapter 3

Visual Encoding Design

Section 3.4 is reprinted from:

R. Sakai, M. Moisse, J. Reumers, and J. Aerts, "Pipit: visualizing functional impacts of structural variations.," *Bioinformatics*, vol. 29, no. 17, pp. 2206–7, Sep. 2013.

Reprinted with permission. License Number: 3691241060484.

3.1 Visual Analytics

The process of mapping information to a visual representation is central to data visualization design. By choosing a graphical representation to encode data, we abstract a value into a visual property. Via analytical reasoning facilitated by visual interfaces (**visual analytics**), we gain a better understanding of the data and underlying patterns, so that we are better positioned to ask more sophisticated questions about the data. A classic example to demonstrate the benefit of visual analytics is Anscombe's quartet. In 1973, the statistician Francis Anscombe constructed four data sets of two variables that have essentially identical summary statistic properties [56]. The data from four tables are visualized as scatter plots by representing each observation as a point on a Cartesian coordinate plane (Figure 3.1). On the scatter plots, the patterns of bivariate data are easy to see and interpret. This example also demonstrates the effect of outliers on statistical properties and limitations of summary statistics.

1		2		3		4	
X	Y	X	Y	X	Y	X	Y
10.00	8.04	10.00	9.14	10.00	7.46	8.00	6.58
8.00	6.95	8.00	8.14	8.00	6.77	8.00	5.76
13.00	7.58	13.00	8.74	13.00	12.74	8.00	7.71
9.00	8.81	9.00	8.77	9.00	7.11	8.00	8.84
11.00	8.33	11.00	9.26	11.00	7.81	8.00	8.47
14.00	9.96	14.00	8.10	14.00	8.84	8.00	7.04
6.00	7.24	6.00	6.13	6.00	6.08	8.00	5.25
4.00	4.26	4.00	3.10	4.00	5.39	19.00	12.50
12.00	10.84	12.00	9.13	12.00	8.15	8.00	5.56
7.00	4.82	7.00	7.26	7.00	6.42	8.00	7.91
5.00	5.68	5.00	4.74	5.00	5.73	8.00	6.89

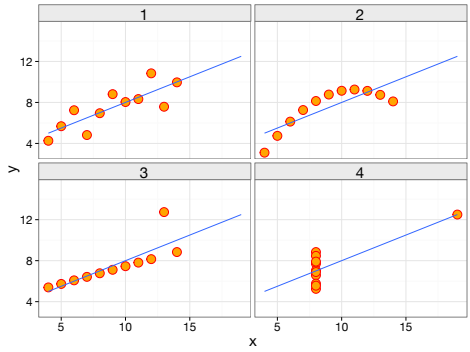

Figure 3.1: The four datasets of Anscombe's Quartet represented in tables and scatter plots.

It should be noted that Anscombe's example does not imply a scatter plot is always a better representation of the data than a table. Whether or how one representation is superior depends on the task of the user (Why), as discussed in Chapter 2. For instance, imagine your task is to recite the exact values of each observation verbatim. You will find the table much easier for this task than reading the position of each dot on the scatter plot. Alternatively, imagine your task is to predict the y value given the x value of 18. The task of prediction is a much more sophisticated cognitive task, requiring the understanding of patterns. With such analysis task, the ability to recite the accurate values is trivial. Instead, the ability to see the pattern and relationships of each observation becomes essential. Thus, the most suitable representation of data depends on the task of the end user.

The design process is about carefully examining different possibilities before making a choice. For example, the utility of the table in Figure 3.1 is improved by just sorting rows based on x values as shown in Figure 3.2. Perhaps the lack of ordering in the original table was intentional to obscure any trends or patterns, but a simple reordering of rows improves the readability and interpretability of the table. Even though sorting is a simple solution, and its impact may be subtle, a design process is all about making these incremental refinements to improve the performance of tasks at hand. Vis in life science requires the thorough exploration of possible visual representations beyond the conventional vis techniques.

As mentioned in Chapter 2, the *Effectiveness Principle* is a fundamental principle of vis design to leverage known facts about human perception to guide the process of choosing the most effective visual marks and channels to translate

1		2		3		4	
X	Y	X	Y	X	Y	X	Y
4.00	4.26	4.00	3.10	4.00	5.39	8.00	6.58
5.00	5.68	5.00	4.74	5.00	5.73	8.00	5.76
6.00	7.24	6.00	6.13	6.00	6.08	8.00	7.71
7.00	4.82	7.00	7.26	7.00	6.42	8.00	8.84
8.00	6.95	8.00	8.14	8.00	6.77	8.00	8.47
9.00	8.81	9.00	8.77	9.00	7.11	8.00	7.04
10.00	8.04	10.00	9.14	10.00	7.46	8.00	5.25
11.00	8.33	11.00	9.26	11.00	7.81	8.00	5.56
12.00	10.84	12.00	9.13	12.00	8.15	8.00	7.91
13.00	7.58	13.00	8.74	13.00	12.74	8.00	6.89
14.00	9.96	14.00	8.10	14.00	8.84	19.00	12.50

Figure 3.2: The reordered Anscombe's Quartet data tables.

data into a graphical representation. A mark is a basic geometric primitive, such as a point, a segment, and an area. A channel is a way to control the appearance of marks, including position, colour, shape, angle and size [7]. The seminal works by Stevens on the psychophysical power law [19] and Cleveland and McGill on rankings of perceptual accuracy for visual channels [20] are often used as guidance to choose a visual channel for the perceptual accuracy based on their data types. Besides the effectiveness criteria, Munzner further elaborates on other considerations on visual encoding in her book: discriminability, separability, and perceptual grouping [7].

Although design principles are useful guidelines, they are not set in stone, and they should not dictate choices you make in the design process. Donald Norman says, "What appears good in principle can sometimes fail when introduced to the world. Sometimes, bad products succeed and good products fail. The world is complex.[57]" Norman is referring to product design in general, but the same applies to the vis design. In the following two design studies, two situations in which the final visual encoding does not follow the *Effectiveness Principle* strictly are discussed.

3.2 Case Study: Fly Plot

Collaboration:

Bang Wong[1] and the Connectivity Map (CMap) research group[1]

B.W. conceived the study and jointly designed the visualization idiom.

[1]*Broad Institute of MIT and Harvard, Cambridge, MA, USA*

The design study on visualization of drug dosage specific response in gene expression, called Fly plot, illustrates the process of visual encoding design and introduces a design strategy for looking beyond the perceptual precision (*Pattern Expressiveness*). The data consisted of expression levels of 1000 genes for about 350 drug compounds at six different dosages measured in 12 different cell lines. Each experiment was replicated. The data was preprocessed into the standard score (z-score) by the domain expert. The gene expression value in response to a drug at a certain dosage in a cell line was expressed as a positive or negative standard score, indicating standard deviations from the control.

The main task for analysis at the point of our involvement was largely exploratory. The domain experts were especially interested in those genes and drugs with dosage responses observed at more than two different dosages. They considered anything above the standard deviations of 2 significant and interesting. In the first design iteration, cell lines were categorised into primary and cancer types. Small multiples of parallel coordinates plot for each selected gene and drug combination were created (Figure 3.3). In each small multiple, the x-axis encodes the drug dosage level, and the y-axis encodes the z-score with thresholds of ± 6. Any value beyond the thresholds was inscribed with a dot. There were two insights from the domain experts' feedback. First, the expert saw a cell-line specific response to a drug, as seen in the *gene_9* and *drug_1*, and the *gene_7* and *drug_4* combinations. Second, the expert also saw some drug response specific to cancer cell lines, as seen in the *gene_4* and *drug_1* combination.

To reduce the number of possible gene and drug combinations, the expert selected about 200 genes with disease annotations out of 1000 genes. Then, we generated a large matrix of small multiples and printed on poster size paper. We found the poster much more inviting and engaging than scrolling a large Portable Document Format (PDF) file on a computer screen when discussing with the domain experts. The medium outside of desktop monitor is further discussed in Chapter 8.

The second iteration (Figure 3.4) involved trying different visual encodings

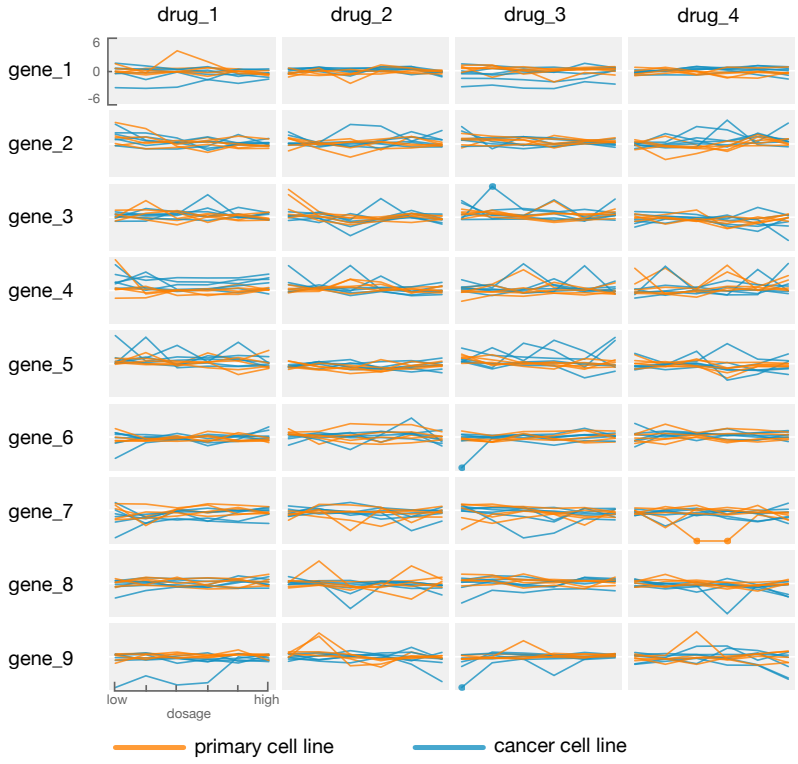

Figure 3.3: The first visual encoding idea. The plot consists of small multiples of parallel coordinates. The cell line types are colour-coded. The position along the x-axis encodes the drug dosage, and the position along the y-axis encodes the z-score value.

to improve the detection of interesting patterns. In this encoding design, the position along the vertical axis encoded the z-score. The area of the circle encoded the dosage level. Each cell line was colour-coded and shown on a separate vertical axis. Any z-score values between the range of 2 and –2 were filtered to reduce the visual noise, because researchers were interested in those patterns that showed a deviation from the midline and measured at more than two different dosages.

Although the use of the position followed the *Effectiveness Principle* and resulted in a very precise and accurate encoding of numerical values, this visual encoding

does neither emphasise nor improve the visual query for the patterns of interest. Those gene and drug combinations that had effects measured at various dosages were represented with overlapping circles with a deviation from the midline. Compared to the previous iteration, the reading of dosage level was less accurate with the area of the circle. Also, as with the first iteration, the visual distinction between neighbouring small multiples was poor. For example, even with a wider border around each small multiple, the lower value range between 0 and –6 was close enough to be mistaken with the upper-value range of the item below. In other words, each small multiple should be more individually distinct so that the viewer would not confuse with the reading of neighbouring small multiples.

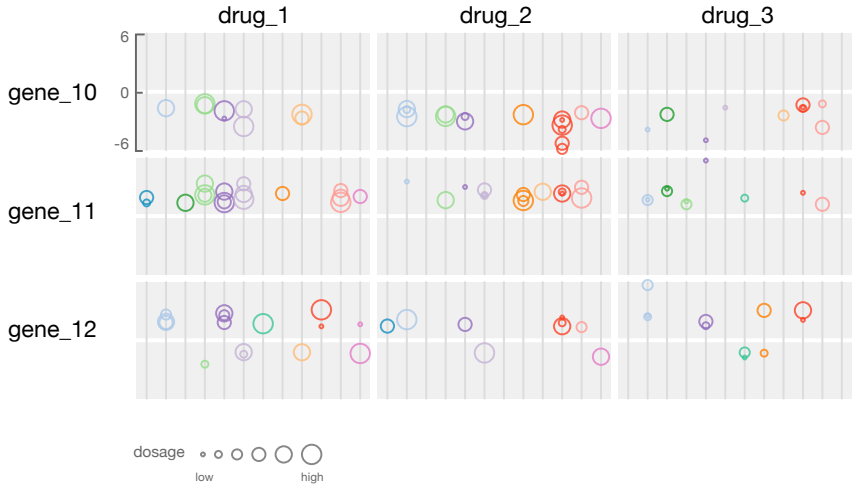

Figure 3.4: The second visual encoding idea. The position along the y-axis encoded the z-score value, and the size of the circle represented the drug dosage level. Each cell line was plotted on a separate vertical axis and colour coded individually. Any scores between 2 and –2 were filtered out.

In the third iteration, we introduced a variant of radar chart, named **Fly plot** (Figure 3.5). The radial axes were ordered in ascending levels of drug dose from left to right and the distance from the centre encoded the value of z-scores (Figure 3.5a). The positive z-scores were shown on the top half of the circle, while the negative z-scores were drawn on the bottom, creating a vertical symmetry (Figure 3.5c). The areas were colour-coded based on the cell line types. With this visual encoding, we expected to see a fan-shaped area for a case of drug effects on expression levels at multiple doses. If there were drug responses specific to a cell line, we expected to see the overlapping fan of

different shapes (Figure 3.5b). The Fly plot uses an existing vis idiom, but we tailored it to this specific dataset to abstract the patterns we expected to see. This process was much like designing an experiment. You customise a "lens" to search for patterns in data, and the design of the lens is driven by the domain specific knowledge. The design strategy to focus on the encoding of patterns is called *Pattern Expressiveness* (discussed in Chapter 2), and the interpretation of these patterns requires experts to integrate their previous knowledge about the function of a gene, the mode of action of a drug, and the characteristic of a cell line.

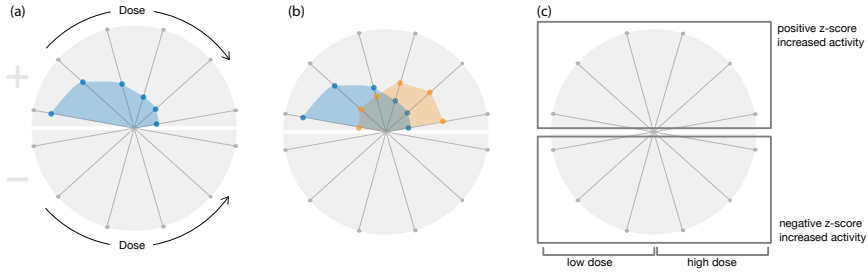

Figure 3.5: Visual encoding of the Fly plot. (a) The positive z-scores are shown on the top, and the negative z-scores are shown on the bottom. The drug dose levels are arranged in ascending order from left to right. (b) An example of expected patterns, where drug effects on expression levels at multiple doses are observed. (c) The vertical symmetry of the plot.

The preliminary results of Fly plot are encouraging. Besides the fan shape that clearly represents dosage specific response to a drug in a gene expression (Figure 3.6a), some unexpected patterns were identified. For example, a gene expression is up-regulated at different doses in different cell types (Figure 3.6b). Another example is a gene modulated at a specific dose range in only cancer cell lines (Figure 3.6c). These unexpected patterns of a gene, a drug and cell lines combinations are valuable insights from the exploratory analysis and still require further investigation to be validated.

To extend the exploratory analysis using Fly plot, we also considered the layout of small multiples. The data structure can be conceptualised as a dataset of three dimensions: genes, drugs and cell lines. The dimension of drug dose was considered as part of the drug dimension since the visual encoding already took different doses into account. This data space can be represented as a cube to consider layout options for small multiples (Figure 3.7a). For example, a user may have a gene of interest to start with, and then the logical choice for the layout is the horizontal slice of the cube, as shown in Figure 3.7b. Likewise, if

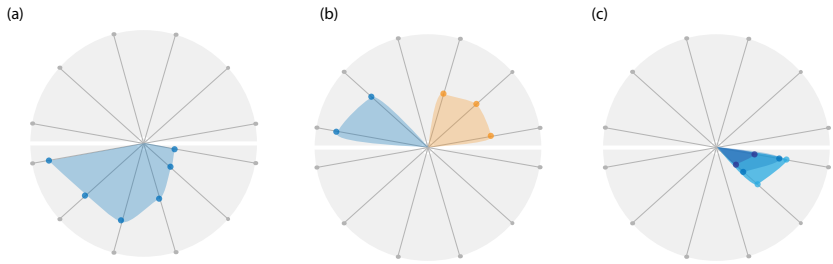

Figure 3.6: Observed patterns in Fly plot analysis. (a) A fan-shaped pattern representing dosage specific response to a drug. (b) A gene expression regulated at different doses in different cell types. (c) A gene modulated at a specific dose range.

a user is studying an effect of a specific drug, or a particular cell line, vertical slices of the cube as in Figure 3.7c, or Figure 3.7d would be more appropriate. Even when using the same vis idiom (How), it is important to consider the task at hand (Why) to choose the most effective layout of graphical representation.

The design study of Fly plot demonstrates that the process of visual encoding design is much like designing an experiment. Most often, you don't know if the choice of visual marks and channels works until you try it with the real datasets. Also, the choice of visual marks and channels is not always about choosing the most perceptually accurate channels according to the *Effectiveness Principle*, but also considering the patterns you expect to see in the data and visually encoding those patterns in the context of the application domain (*Pattern Expressiveness*). This study is still in progress, and the subsequent extension of Fly plot integrates computational approaches, such as the Self Organising Map (SOM). Computational approaches in vis design are discussed in Chapter 7.

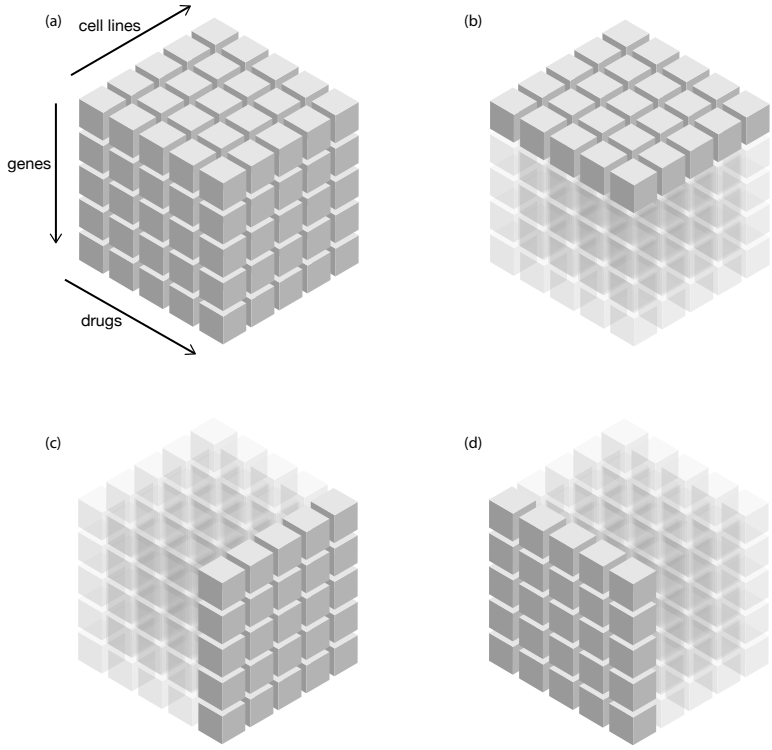

Figure 3.7: Conceptual representation of the gene modulation dataset as a cube. (a) Dimensions of this cube are genes, drugs and cell lines. (b) Examining a gene against all drugs and cell lines. (c) Examining a drug against all genes and cell lines. (d) Examining a cell line against all genes and drugs.

3.3 Case Study: Pipit

Collaboration:

Matthieu Mosisse[1,2] and Joke Reumers[3]

M.M. and J.R. were domain experts and target users. J.R. supervised the project.

[1] *Vesalius Research Center, VIB and KU Leuven, Leuven 3000, Belgium*
[2] *Laboratory of Translational Genetics, Department of Oncology, KU Leuven, Leuven 3000, Belgium*
[3] *Janssen Infectious Diseases–Diagnostics, Janssen Pharmaceuticals, Beerse 2340, Belgium*

In collaboration with computational biologists studying genomic structural variations of cancer, we developed an interactive visualization tool, called **Pipit**. We introduced a novel visual encoding to abstract the genomic size and distance of structural variations and to focus on the functional impacts of large or small structural variations. This design choice was motivated by the fact that existing visual encodings tend to focus on the size and position, rather than the potential consequence of structural rearrangements. Small structural variations may have more severe effects than large variation, but it may be easily overlooked with existing visual encodings. The data in genomics are full of semantics, and sometimes advances in vis design come from re-evaluating the semantics to focus on. In this design study, we chose to focus on the underlying gene structures rather than the genomic distance.

The task (Why) and the vis idiom (How) of Pipit are clear in retrospect, but the actual design process was not straightforward, involving three iterations before reaching the final design. When we started the collaboration, researchers were analysing structural variations of uterine cancer. The whole genome sequence of a tumour sample and a healthy tissue sample were processed, and structural rearrangements were detected by an algorithm based on the positions and orientations of paired-end reads [51]. Four examples of structural variation types and associated read-pair mappings are shown in Figure 3.8.

Even with a support of schematic illustrations (Figure 3.8), understanding the underlying structural rearrangements from paired-end reads is not easy, and it takes some practice and experience. During the interview, the domain expert shared his notes where he started to sketch a complex structural rearrangement region where many segments between breakpoints were overlapping (Figure 3.9). Especially in the early stages of development, these visual clues found in the

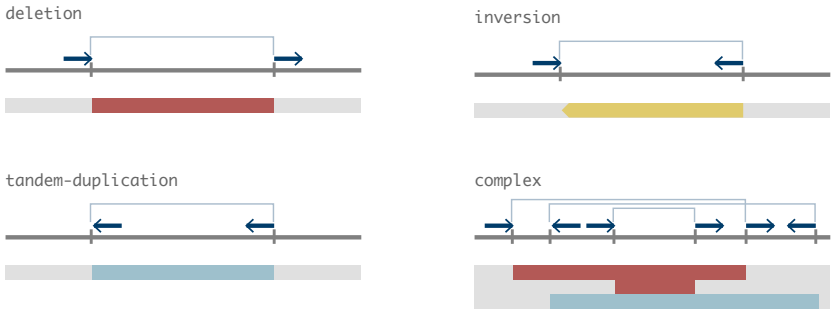

Figure 3.8: Schematic illustration of structural rearrangements inferred from the paired-end reads and mate gaps. The navy line segments represent reads connected to another.

domain experts's notes are quite useful to understand the challenges they face and to inform the design for prototyping. As shown in Figure 3.9, the expert drew all the paired-end reads and the reads' orientations. On the top right, the expert started to decode, but the task quickly became too tedious and the visual representation became cluttered, so he gave up. After a few interviews with domain experts, the target task for the first prototype was defined. The objective was to explore the detected structural rearrangements and show the associated paired-end reads positions and orientations to verify the performance of the algorithm.

With sample datasets of structural rearrangement annotations and the paired-end reads information, the *first prototype* was developed (Figure 3.10). The grey bar at the top represents a whole length of a chromosome, and chromosome 1 is shown in this view. Vertical lines within the grey bar indicate breakpoints. Inter-chromosomal rearrangements are indicated with labels of the mapped chromosome. The section below shows a zoomed region with paired-end reads and inferred structural rearrangements that are colour coded by their rearrangement type. The text below shows a list of input data points. The interaction is limited to the left and right button on the keyboard to switch to the previous or next structural rearrangement events.

The user interface and interaction may be simple and crude, but the goal of a prototype is an early delivery of a concept and the speed of development is more important than polish. The *first prototype* allowed the researcher to validate their structural rearrangement detection algorithm and prompted subsequent research questions. While testing the prototype, the researcher expressed that

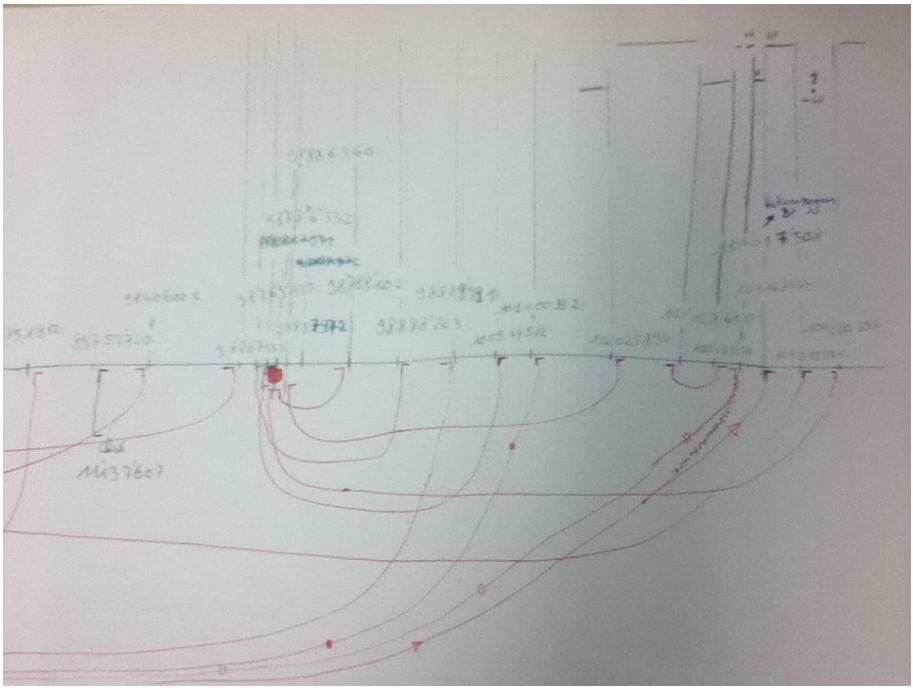

Figure 3.9: A photograph of an expert's note. The red line shows the paired-end read. The positions and the read orientations are indicated. This page is one of the three-page long sketch the expert drew.

they wanted to compare the tumour sample against the control and investigate affected genes with respect to structural rearrangement events. These insights into a vis tool development may be obvious and logical in retrospect, but often eliciting these requirements and tasks is not easy without a prototype. Through iterations, the target task (Why) is refined using prototypes. Prototypes are low-cost in terms of time and effort, and especially suitable for transitional and iterative design process.

The *second prototype* compared two samples and included a filter function to hide the common structural variation events detected in both the tumour sample and the control sample (Figure 3.11). This prototype had additional layers of information. The gene tracks with their gene symbols were dynamically drawn with respect to the selected region. In Figure 3.11, a region of chromosome 4 is selected to investigate the large structural variation present in the tumour sample. The next level showed that the tumour sample has a distal duplication and a deletion event overlapping with each other. When a user selected the deletion

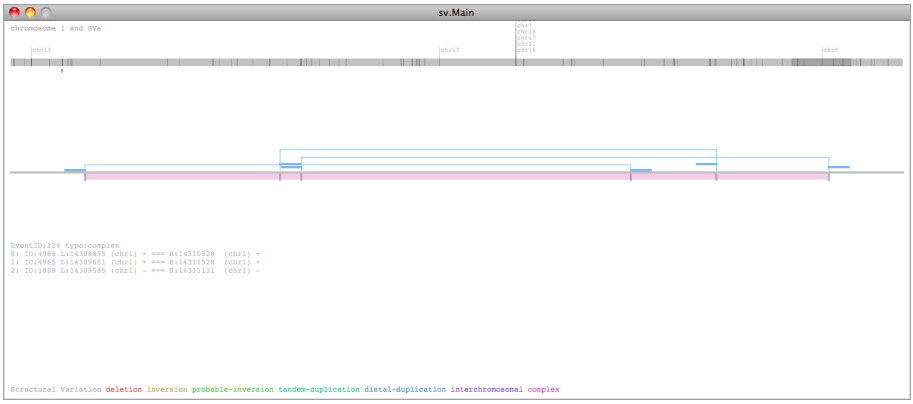

Figure 3.10: The interface of the first prototype. The grey bar at the top represents a chromosome and breakpoints of structural variations are indicated within the bar. Inter-chromosomal rearrangements are indicated with the mapped chromosome name. The middle section shows a selected structural rearrangement event. It showed paired-end reads and inferred structural rearrangement that are colour coded. The image shows a complex structural rearrangement event.

event, the prototype showed the underlying gene track underneath. One useful outcome of using this version of the prototype was that experts identified more structural variations events in the control sample than they expected, especially after filtering out those common structural rearrangement events. They also noticed very large structural variations in the control cases. These insights provided some clues to revisit their algorithms and reinvestigate the sequencing quality of the specific regions to validate these structural rearrangement events.

Especially in the early stages of development, it is important to explore other design options. The *third prototype* used a circular layout, like the Circos vis idiom [58] (Figure 3.12). In this view, the deletion (blue), tandem-duplication (orange) and inter-chromosomal rearrangements (green) were visualized. Advantages of this layout were that it provided an overview of the whole genome and the inter- and intra-chromosomal rearrangements were much easier to see. In this prototype, semantic zooming was implemented to reveal more detailed information, such as the cytoband annotation, while maintaining its context (Figure 3.13). Although this prototype was not further developed, it provided some interesting visualization concepts. For example, the circular layout smoothly transitioned to a linear representation by zooming in. Typically, a Circos plot is static, and this type of interaction design can support focus plus context analysis [7].

With these prototypes, the user tasks were re-evaluated, and the objective of

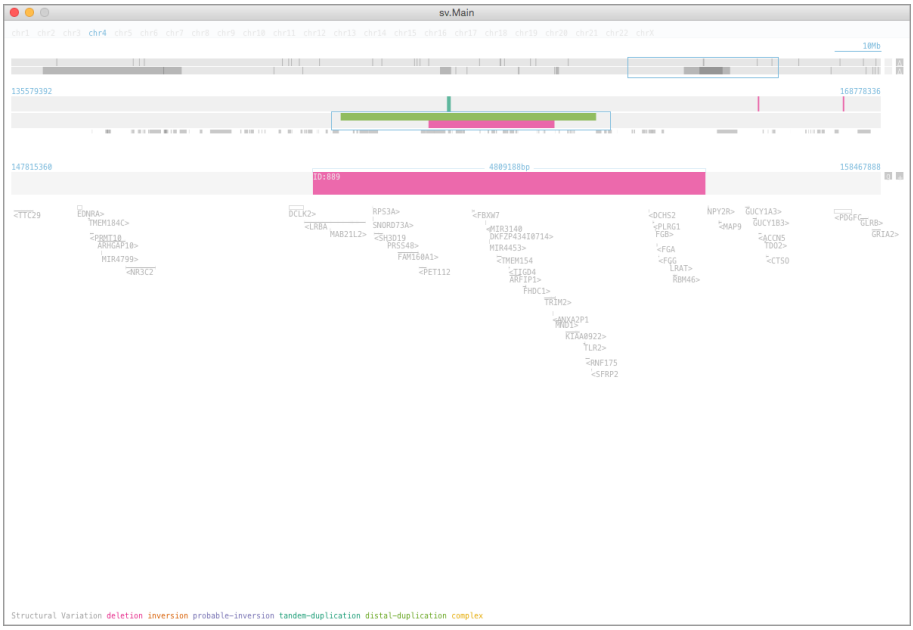

Figure 3.11: The interface of the second prototype. A segment of chromosome 4 in the tumour sample is selected.

the vis deign was redefined to the characterisation of the functional impact of structural rearrangements. For instance, instead of just listing gene symbols (Figure 3.11), Gene Ontology (GO) terms can be integrated to infer the potential impact of affected genes. At this point, the card sorting technique (described in Chapter 2) was employed to re-evaluate the tasks (Why) and the goal of the vis tool. The results of this exercise encouraged us to focus on the metadata, such as GO terms and haploinsufficiency scores [59], instead of genomic positions and distance. As results, we developed a novel visual analytics tool, named Pipit, to focus on the characterisation of the functional impact of structural rearrangements.

Pipit uses a visual encoding where each affected gene is represented as a circle. The fill pattern is based on the affected part of the transcript, and the fill colour is based on the type of rearrangement (Figure 3.14). Unaffected genes are simply compressed into a line connecting affected genes. The tool provides four different layout options: collapsed, expanded, chromosomal position and unit plot views (the details of these views are discussed in the paper attached at the end of this chapter). In Figure 3.15, structural variations of the mouse genome are shown with annotations of known oncogenes and GO terms. In

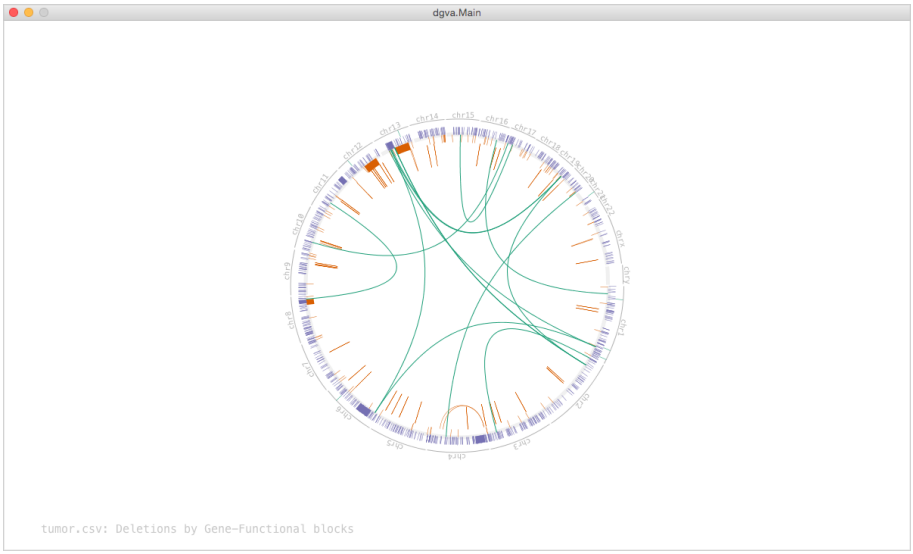

Figure 3.12: The interface of a prototype based on a circular layout.

this view, the deletion of the *Met* gene is selected, and GO terms associated with this gene are listed in the right panel. The bottom panel incorporated ideas from the previous prototypes, showing the genomic region of the gene and different transcripts associated with the selected gene.

By changing to a gene-centric analysis approach, the entry point of the exploratory research question, and the focus of visualization tool changed to leveraging the previous knowledge about the affected genes. For example, a researcher may be interested in checking the haploinsufficiency of genes affected by structural rearrangements. The user can input a list of genes with haploinsufficiency scores and Pipit matches the list against affected genes. In Figure 3.16, the affected genes with haploinsufficiency score above 0.8 are highlighted with a black outline. When the user selected the circle highlighted in magenta, three genes (*RRP8*, *ILK*, and *TAF10*) consecutively located and affected by deletions events were shown on the bottom. By switching from the collapsed view to the expanded view, the user can examine these genes individually, as shown in Figure 3.17. It shows that the *TAF10* gene is the gene with a high haploinsufficiency score, and the associated GO terms are updated on the right panel.

The design study of Pipit demonstrates a case where prototypes defined the target task and led to a novel visual encoding design. The choice of visual marks and channels was not based on *Effectiveness Principle*. The key design

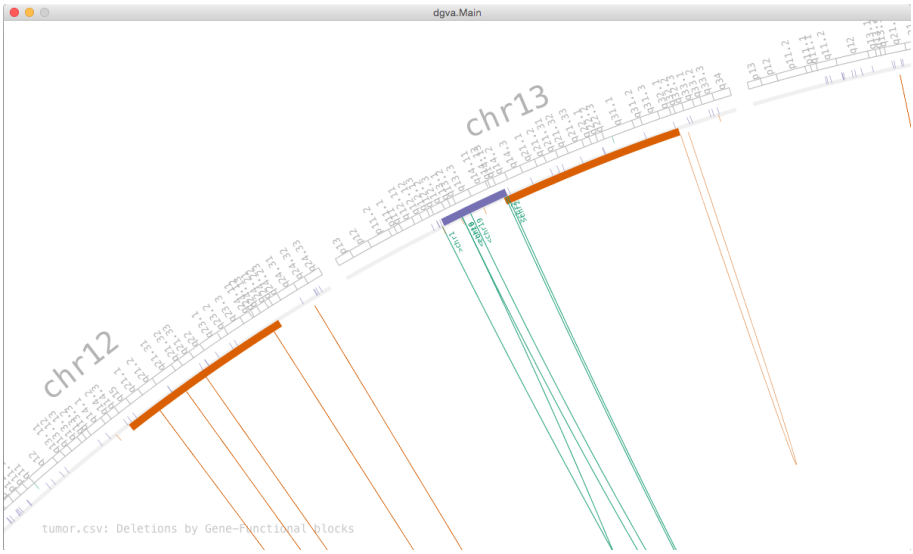

Figure 3.13: The interface of a prototype showing a zoomed region. The prototype included a semantic zooming function, where more detailed information was revealed at a zoomed level.

strategy was the abstraction of the genomic position and size of structural rearrangements in order to focus on the characterisation of the functional impacts by incorporating external datasets. The details of the application and use cases are described in Section 3.4.

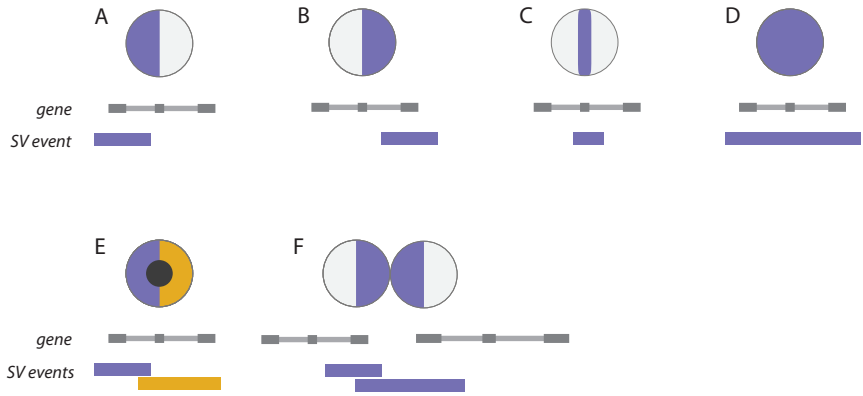

Figure 3.14: Visual encodings of structural variations (SV) in Pipit. (A) The left side of the gene is affected by an SV event. (B) The right side of the gene is affected. (C) The middle portion of the gene is affected. (D) The entire gene is affected by an SV event. (E) More than two SV events overlap is denoted with a dot in the middle of the circle. (F) A representation of a potential fusion gene with deletion events.

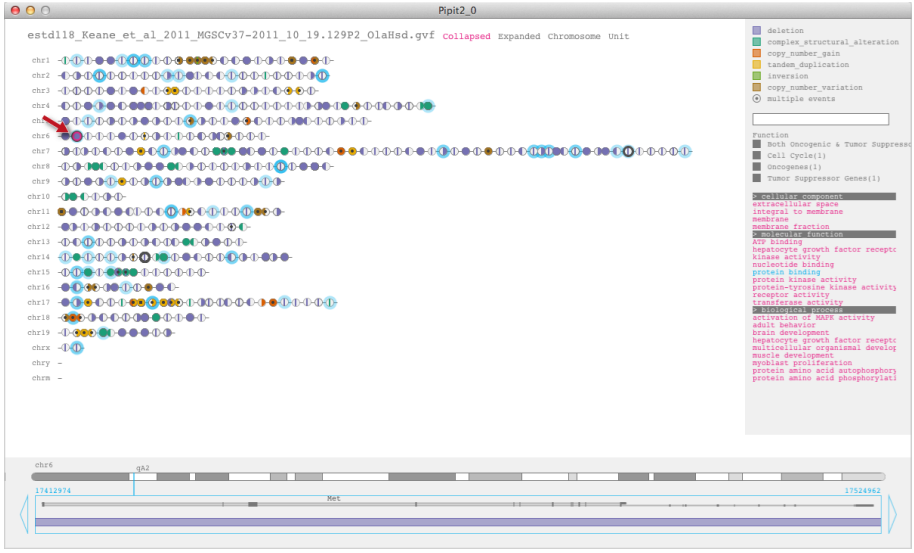

Figure 3.15: The interface of Pipit. The view shows structural rearrangement of the mouse genome in the collapsed view. Affected genes that are associated with "protein binding" GO term are highlighted in light blue.

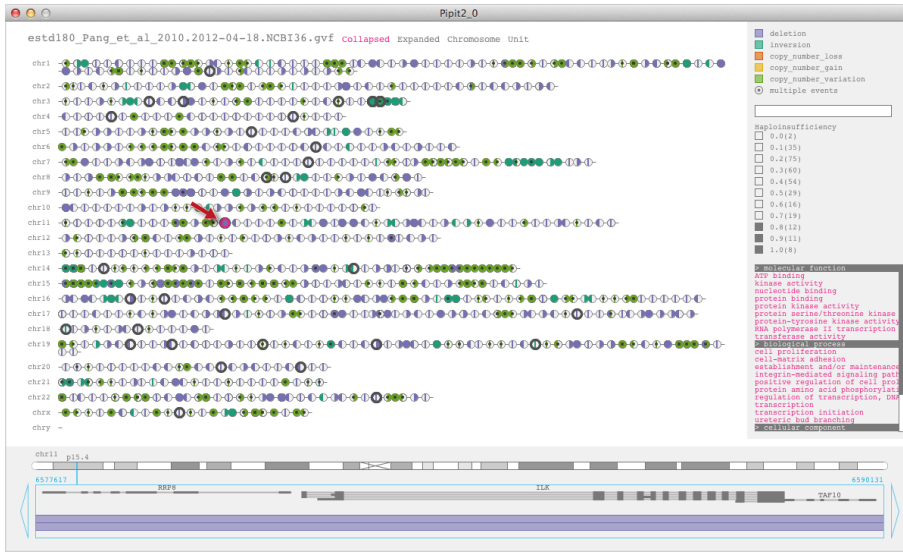

Figure 3.16: The user interface of Pipit in the collapsed view. The affected genes with haploinsufficiency score above 0.8 are highlighted with a black outline.

Figure 3.17: The user interface of Pipit in the expanded view. The *TAF10* gene is the gene with a high haploinsufficiency score. The associated Go terms are updated on the right panel.

3.4 Pipit: Visualizing Functional Impacts of Structural Variations

R. Sakai, M. Moisse, J. Reumers, and J. Aerts, "Pipit: visualizing functional impacts of structural variations.," *Bioinformatics*, vol. 29, no. 17, pp. 2206–7, Sep. 2013. (Reprinted with permission. License Number: 3691241060484.)

3.4.1 Summary

Pipit is a gene-centric interactive visualization tool designed to study structural genomic variations. Through focusing on individual genes as the functional unit, researchers are able to study and generate hypotheses on the biological impact of different structural variations, for instance the deletion of dosage-sensitive genes or the formation of fusion genes. Pipit is a cross-platform Java application that visualises structural variation data from Genome Variation Format (GVF) files.

3.4.2 Availability:

Executables, source code, sample data, documentation and screencast are available at https://bitbucket.org/vda-lab/pipit/. Supplementary materials are available at Bioinformatics online [54].

3.4.3 Introduction

Structural variation is defined as a change of genomic DNA greater than one kilobase in size and can be either balanced or unbalanced [60]. A structural variation may be benign, it may influence phenotypes, it may predispose to or cause diseases, and it may be transmitted to next generations. In addition, it may also result in the formation of new transcripts through gene fusion or exon skipping when breakpoints disrupt gene structures [61]. Understanding the structural change in the genome, as well as its functional impact, is critical for studying phenotypic variations and genetic diseases in human and model organisms [62].

When structural variations are studied, the structure of these variants are usually visualised by encoding breakpoints on a linear or circular layout [58, 63]. Other visual encodings such as dot plot and graph representations show the

changes in the reference genome introduced by the structural rearrangement [64]. These visual encodings pose two challenges in analysis of structural variation data. Firstly, they focus on size and position, rather than the consequence of structural rearrangement. Secondly, they introduce an implicit correlation between size and effect, while small structural variations may have more severe effects than some large variations. Therefore, an effective visual encoding of structural and copy-number variations based on functional units is essential to gain insights into impact on human health and disease.

We introduce an exploratory visualization tool, named Pipit, which utilises a novel gene-centric visual encoding to examine how gene structures individually are affected by structural variants. Pipit takes a Genome Variation Format (GVF) file as input, and encodes an affected gene structure as a disk, showing how it is modified and oriented towards other genes. By means of this data abstraction, the focus of the structural variants is shifted from the genomic distance to the biologically relevant feature, i.e. the underlying genes. This encoding enables swift visual inspection of genes that are affected by structural variants, which may be examined further if deemed interesting. Not constrained or biased by variation sizes, Pipit permits in-depth analysis and exploration of structural variation data.

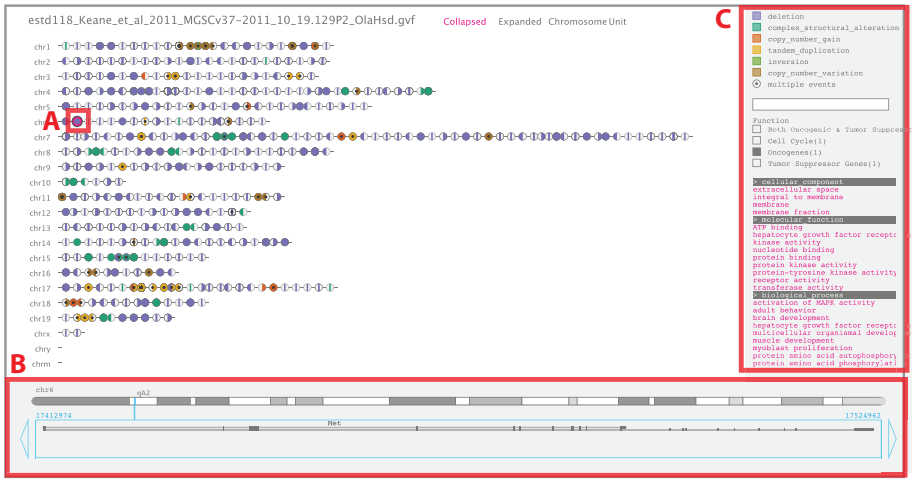

Figure 3.18: Pipit visualising the structural variation data of the mouse genome (estd118). (A) The deletion of the *Met* gene on chromosome 6 is selected. (B) The associated genomic information of the region is shown on the bottom panel. (C) Categories of the known oncogenes for the mouse are listed and the *Oncogenes* category is selected. Gene ontology terms associated with affected genes are listed below on the right panel.

3.4.4 Features

Pipit is an interactive visualization tool developed in Processing, an open source programming language and integrated development environment (IDE) based on Java. The executables are available for Linux, Mac OS X and Windows. The input file is in GVF because it is a well standardised format for genomic structural variation data, extended from the Generic Feature Format (GFF3) [65], and both the European Bioinformatics Institute (EBI) and the National Center for Biotechnology Information (NCBI) curate, archive, and make data publicly available in this format via DGVa (http://www.ebi.ac.uk/dgva) and dbVar (http:// www.ncbi.nlm.nih.gov/dbvar) respectively [66].

Pipit also uses the gene track, cytoband and gene ontology information obtained from the UCSC table browser database [67]. The current version supports the data from human (NCBI build 36 and 37) and mouse (NCBI build 37/mm9 and GRCm build 38/mm10), but the user can load the data for other model organisms. In addition, the user can load a comma-separated values (CSV) file containing the Ensembl gene ID and ordinal or categorical information, such as haploinsufficiency scores [59] and known oncogenes (Figure 3.18 and Supplementary Material [54]).

Each affected gene is represented as a disk and filled according to which part of its structure is influenced by a structural variation (Figure 3.18. See Supplementary Material [54]). Structural variant types are based on the data file and colour coded as shown on the right panel. Unaffected genes are compressed into a line connecting affected genes. The default promoter length upstream of the gene sequence can be set when loading the data.

There are four layouts to explore the structural variation data. The default view is the *collapsed, ordered gene view* (Figure 3.18). In this view, a coloured disk may represent an affected gene or consecutively ordered genes that are affected by the same type of structural variation. In the *expanded view*, all affected genes are individually visualised. The *chromosome position view* shows affected variants mapped to their genomic positions. Lastly, the *unit plot view* visualises affected genes by their type of structural variant event, such as deletion, tandem-duplication and so forth (see Supplementary Material [54]).

When a disk unit is selected (Figure 3.18A), the underlying genes and structural variation events are shown on the bottom panel, along with the chromosome with cytobands and transcripts with their exonic regions coloured in dark grey (Figure 3.18B). The gene name shown in this panel links to the Ensembl browser and displays the genomic region. In the right panel (Figure 3.18C), the coloured square boxes for each structural variant types serve as radio buttons to hide or show a selected type of variant. The text field below searches for a specific gene

amongst affected genes. Gene Ontology (GO) terms associated with affected genes are listed, and conversely selecting a GO term highlights associated genes in the main view. A screenshot can be saved as a Portable Document Format (PDF) by pressing the ‘p‘ on the keyboard.

3.4.5 Discussion

Pipit introduces a novel visualization paradigm and user interaction method to examine structural variants based on the affected gene region. It facilitates the study of structural variants from a gene-centric perspective to investigate various events, for instance, how known dosage-sensitive genes are affected or whether gene fusions are formed. Future work includes extending the functional unit to encompass important regulatory elements, as elaborated in the ENCODE project [68]. Additionally, functions to compare multiple samples from various data formats, such as Variant Call Format [69], and options to incorporate other conventional linear or circular representations are essential for more comprehensive study of structural variants.

3.4.6 Acknowledgement

Funding This work was supported by iMinds [SBO 2012], University of Leuven Research Council [SymBioSys PFV/10/016, GOA/10/009] and European Union Framework Programme 7 [HEALTH-F2-2008-223040 CHeartED].

Conflict of Interest none declared.

Chapter 4

Data Sketching

Section 4.3 is reprinted from:

R. Sakai and J. Aerts, "Sequence Diversity Diagram for comparative analysis of multiple sequence alignments," *BMC Proc.*, vol. 8, no. Suppl 2, p. S9, 2014.

R. Sakai and J. Aerts, "Erratum to : Sequence Diversity Diagram for comparative analysis of multiple sequence alignments," *BMC Proc.*, vol. 8, no. Suppl 2, p. S10, 2014.

4.1 Why Data Sketch?

In a data visualization design process, a designer makes a series of choices, two of which we will highlight here. The first type of choice is about *selecting the variables to draw*. You may decide to select certain attributes, or you may transform the data to derive new variables to draw. For example, you may choose two continuous variables to draw a scatter plot from a multidimensional table. Or, you may choose to perform Principal Component Analysis (PCA) to reduce dimensions and use the principal components to draw a scatter plot. Choosing which variables to draw is central to the "What" question of the What-How-Why framework. The second type of choice is about *mapping of data points to visual marks and channels*, which relates to the "How" question. This mapping process is a translation of a value, be it quantitative, ordinal

or nominal, to a graphical representation. Because the design space of vis design is so large and most of many encodings are ineffective [7], the challenge in visualization design is to explore options and evaluate their effectiveness efficiently.

In this chapter, we introduce the concept of **data sketching** for static data vis design for the abstract data type. There are three fundamental concepts of data sketching: 1) the use of real datasets, 2) the iterative design process, and 3) the speed over polish. First, the use of real datasets from the beginning is an important tactic for a number of reasons. Common design study pitfalls include cases where the toy data is inappropriate, the delivery of real data is delayed, and the dataset is simply inadequate [6]. Second, the design process should be an iterative refinement, instead of it being one single design cycle. This iterative approach prevents tunnel vision, and instead it encourages the designer to consider possible design options as widely as possible. Lastly, as the term "sketch" implies, each iteration should emphasise speed over polish. This method is referred to as computer-based low-fidelity prototyping in Human-Computer Interaction (HCI). The resulting representation may not be at the publication-ready quality, but it is good enough to evaluate whether your vis idiom is effective for the task.

The goal of data sketching is to try out different visual encodings and data transformations to optimise the design for the intended analysis tasks. The frame of mind for this process relates to a slogan used by one of the IDEO groups, "Fail often, and fail fast" [57]. In other words, a designer should explore as many ideas as possible before committing their time and effort to engineering one design solution. In the following, the data sketching process is described.

For the exercise of data sketching, we advocate tools such as Processing [70]. As mentioned in Chapter 2, there are several visualization tools and each has its pros and cons. The choice of tool is up to the designer, as long as they can satisfy the three data sketching concepts mentioned above. For example, Giorgia Lupi and Stefanie Posavec literally sketch the data they collect from their everyday life [71]. Their visualizations in a postcard format are very expressive. On the other hand, the vis process without any programming can be laborious and tedious at times as they also acknowledge this aspect in their presentation [72]. Other visualization tools, such as Excel or Tableau, are suitable for quickly visualizing and choosing the variables of interest, but the scope of visual design is often restricted by the template-based visualization approach. Processing, on the other hand, does require some programming skills, but it makes drawing something on a computer screen accessible to achieve very expressive visualization designs.

Besides the low learning curve, there are three advantages of using Processing

for development. The first advantage of Processing is its extensions. Processing is based on Java, and you can integrate any existing Java libraries. This feature gives flexibility to the development and options to extend functionalities to filter, aggregate or transform the data. The second advantage is its modularity. Using the object-oriented programming approach decouples the model and the view. This model-view-control architecture allows iterative refinement of visual encoding and the data structure independently. Also, we often encountered cases where a collaborator asks to add new data sets. In these situations, it is relatively easy to parse the new data into another object added to the existing data structure. The third advantage is that the philosophy of small-scale development style behind Processing resonates with the data sketching concepts. In Processing, each program is called a "sketch", adopting the process of scripting to write code quickly [73].

In the following, the design process of Sequence Diversity Diagram (SeDD) is presented as a case study of data sketching. We elaborate on the design process of SeDD, including the descriptive detail of the intermediate steps leading to the final design. At the end of the chapter, a publication on this visualization is attached. The article includes the scientific background and the evaluation of a use case.

4.2 Sequence Diversity Diagram - BioVis Redesign Challenge

The BioVis 2013 conference posed the challenge of redesigning the sequence logo. The sequence logo has been a long-standing convention for visualization of multiple sequence alignments of nucleotides or amino acids for the past few decades [74]. Although the sequence logo is an effective representation of motifs, this vis idiom falls short when the main analysis task is to compare between two or more sets of aligned sequences. The contest provided figures of sequence logos along with the figure legend and the input datasets. The dataset consisted of aligned amino acid sequences of the adenylate kinase lid (AKL) domain of gram-positive and gram-negative bacteria. The provided figure compared sequence logos across all organisms (Figure 4.1A), from gram-negative bacteria (Figure 4.1B) and gram-positive bacteria (Figure 4.1C). The figure legend provided some context and shed some light on the user's intentions and underlying tasks.

The redesign process starts by examining the strengths and weaknesses of the existing representation with respect to its intended tasks. Even without considering its scientific context, the use of a primary, saturated colour scheme

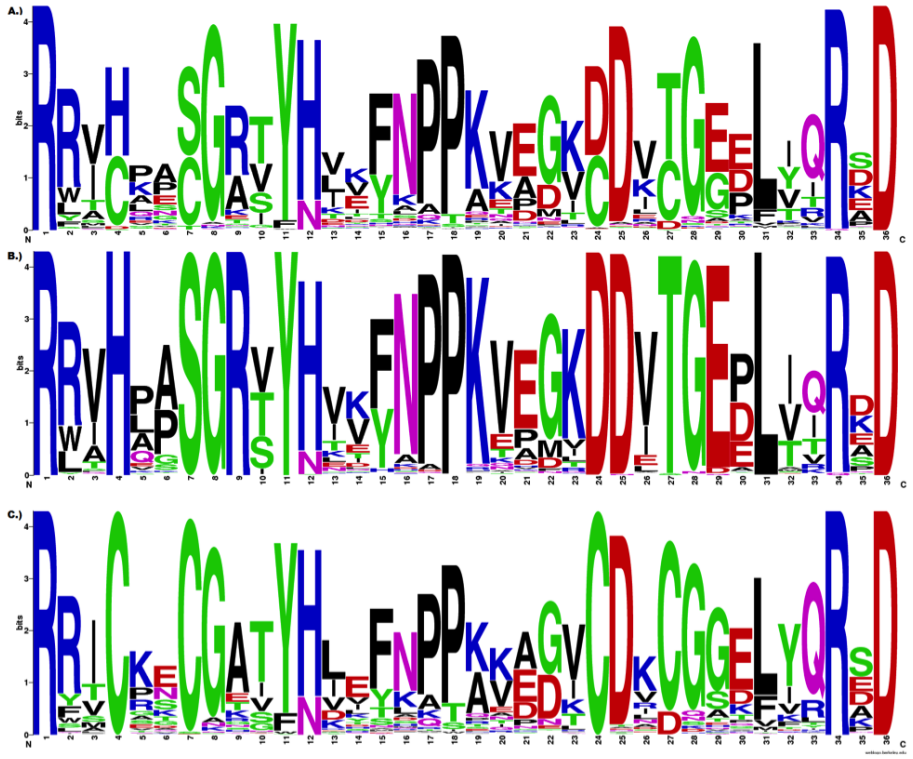

Figure 4.1: Sequence logos of the adenylate kinase lid (AKL) domain. (A) Across all organism. (B) Gram-negative bacteria. (C) Gram-positive bacteria. The ADK lid domain structure is universally conserved, but is stabilized in the Gram-negatives by a hydrogen bonding network between residues 4, 7, 9, 24, 27, and 29 (and several other residues in some organisms), while the Gram-positives are stabilized by a bound metal ion, tetrahedrally coordinated by the Cysteines at 4, 7, 24 and 27. The identities of several other positions (e.g. 5, 8, 30, 32) are differentially constrained in each subfamily as well, apparently due to steric requirements of the stabilizing residues. (This figure and the dataset were provided by Drs. Magliery and Sullivan at the Ohio State University for the purposes of the BioVis 2013 Contest.)

is incredibly distracting, especially when you try to compare between sequence logos. In this vis idiom, colour encodes categorical information of the structural and chemical properties of the amino acid residue. Because the functional group annotation is only implicit and is not even mentioned in the figure legend, this information is probably not the most important message of the figure, nevertheless, the use of a bold colour scheme makes the functional grouping the most perceptually dominant element of this figure (*Pop-out Effect*).

The legend of the provided figure (Figure 4.1) suggests that the main analysis tasks are to compare between multiple sequence alignment sets and to compare conserved positions. In the sequence logo vis idiom, the total height for each position encodes the information content, also known as the Shannon entropy [75]. The total height is subdivided based on the frequency of each amino acid, and a letter representing the amino acid is scaled accordingly. Hence, this representation emphasises the positional conservation, and it is ideal for representing a motif. However, its downside is that it is not possible to infer the sequential conservation of two consecutive positions. For example, it is not clear whether the Proline (P) at position 5 is followed by an Alanine (A) or another Proline (P) at position 6 in gram-negative bacteria (Figure 4.1B). Hence, the initial motivation for the redesign was to improve the comparison of sequential conservation of adjacent positions.

If the main objective is to provide a quick overview and to enable rapid visual scanning for most conserved positions, the sequence logo works perfectly as a motif. For example, the use of sequence logos as motifs is a well-justified method for visualizing transcription factor binding sites or functional motifs in amino acid sequences. However, for tasks of comparing the frequency between different alignment sets, the vertical juxtaposition of two figures does not help to compare the height of the letters and any subtle differences are difficult to spot.

In order to understand the information theory and computation behind the sequence logo vis idiom, the sequence logo representation was roughly reproduced (Figure 4.2). This rough data sketch served as a starting point for evaluating its pros and cons.

In the sequence logo, each letter height is proportional given the total height based on the degree of certainty at that position, hence the comparison of frequencies of amino acids between two positions is not supported. In other words, the frequency cannot be directly compared between two positions, unless they have the same total height. Figure 4.3 shows a design iteration to simplify the sequence logo by just encoding the frequency of amino acids at each position. The height of a bar represents its frequency, and the order of residues at each position was sorted based on the frequency. In this representation, the letters of less frequent amino acid residues are illegible, thus gaps between bar segments are introduced in Figure 4.4. While the less conserved positions have a shorter total height in the sequence logo, those positions have a greater total height because of gaps introduced. Without the use of the information content, the visual emphasis or salience on inferred conservation is reduced, but the frequency between positions becomes directly comparable.

To encode the information about sequential conservation between two adjacent

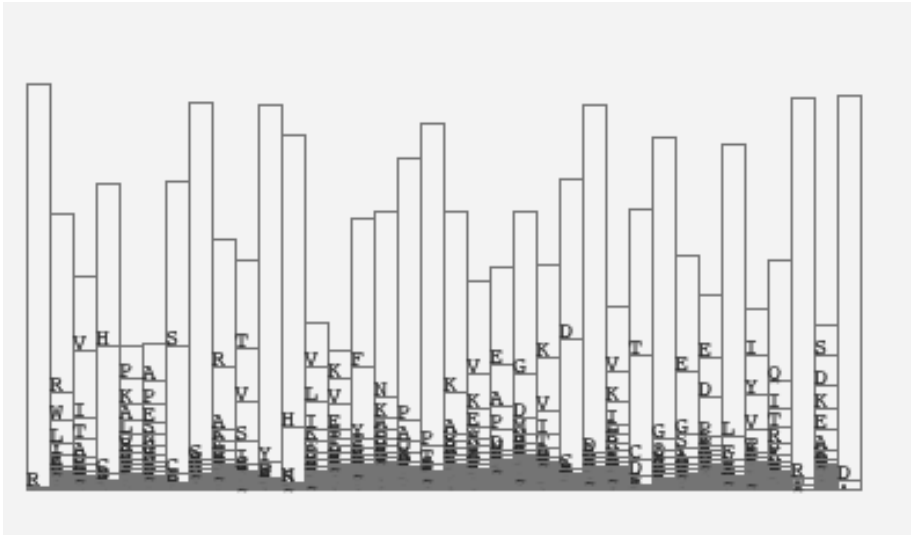

Figure 4.2: The first data sketch reproducing the sequence logo technique.

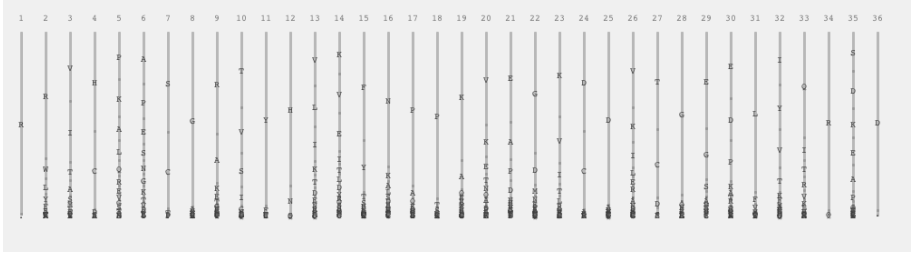

Figure 4.3: The data sketch of frequencies at each position.

positions, lines with varying thickness were drawn to encode the frequency of two consecutive residues (Figure 4.5). With this data sketch, it became clear that there are some sequential patterns; there are thick lines as well as very thin lines. This vis idiom is essentially Parallel Sets [76]. One key difference is the underlying data structure; the aggregated multiple sequence alignment (MSA) data is a graph structure, instead of a tree structure. Figure 4.6 shows a variation where the frequency is encoded as the area of circles, which results in a more compact representation. With discrete circles, it is easier to scan visually for conserved positions. This iteration step demonstrated that the choice of visual marks influences the emphasis on different information. Also, trying different visual encodings was made easy because of the decoupling of

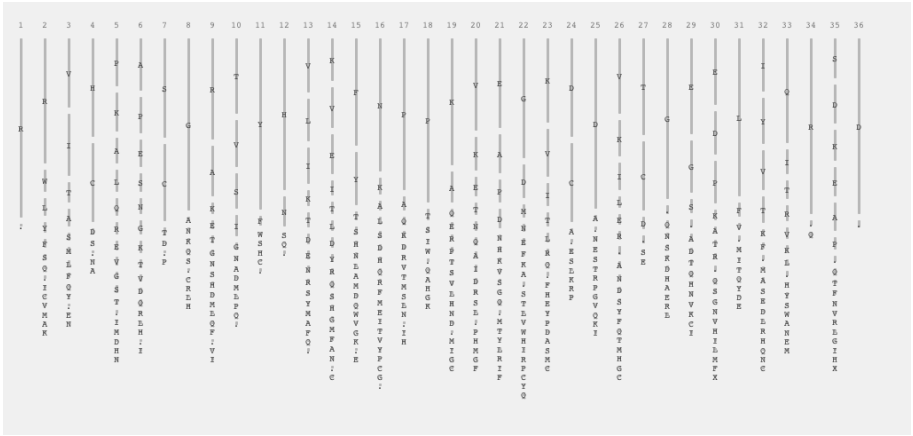

Figure 4.4: The data sketch of frequencies at each position with gaps in between.

the model and the view.

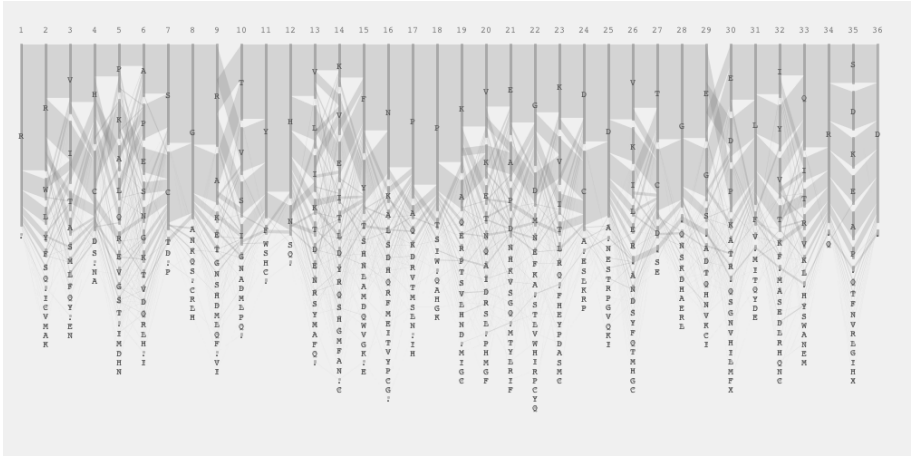

Figure 4.5: The data sketch to include the sequential conservation information.

Until this point, only one set of sequence alignment data was used for data sketching for simplicity. The following steps focused on the comparison of two sets of sequence alignments as the main analysis task. Figure 4.7 extends the visual encoding in Figure 4.6. The proportion of two sets was compared using the pie chart vis idiom. This visual encoding supports visual queries for comparative analysis tasks. For example, positions that differentiate between two sets were identified by scanning for two medium-size circles of different

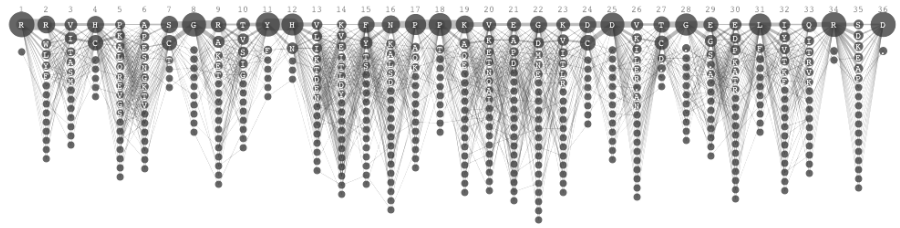

Figure 4.6: A variation of Figure 4.5, where the area of the circle is used to encode the frequency.

colours, such as position 4, 7, 24, and 27. On the other hand, positions that are common in both sets are identified by scanning for large circles with a division, such as position 1, 11, 34, and 36. While the pie charts compare the proportion at each position, it does not convey any sequential patterns.

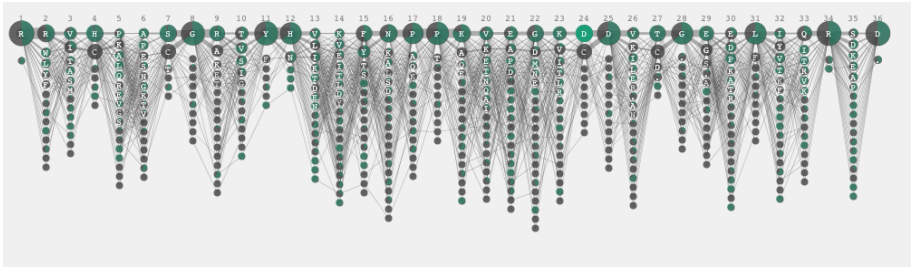

Figure 4.7: A data sketch with pie charts to compare proportions of two alignment sets per residue.

In the subsequent iteration, the previous parallel set encoding (Figure 4.5) was revisited to add the sequential conservation between two positions (Figure 4.8). The first vertical axis shows the frequency of gram positive and negative alignment sets. The gram-negative sample set is highlighted in magenta. There were some issues in this implementation. For instance, some lines were not continuous due to some bugs in the code for sorting subsets. Also, the identity of amino acid were not shown in this view. The data sketching process involves these iterative testing and evaluation of pros and cons for each visual encoding idea.

In the data sketching process, you strive for a sweet spot where the choice of visual encoding balances the pros and cons for intended tasks. Through iteration, you apply lessons learned from previous ones and try new ideas on a trial and error basis. Sometimes you go back and forth between ideas. The

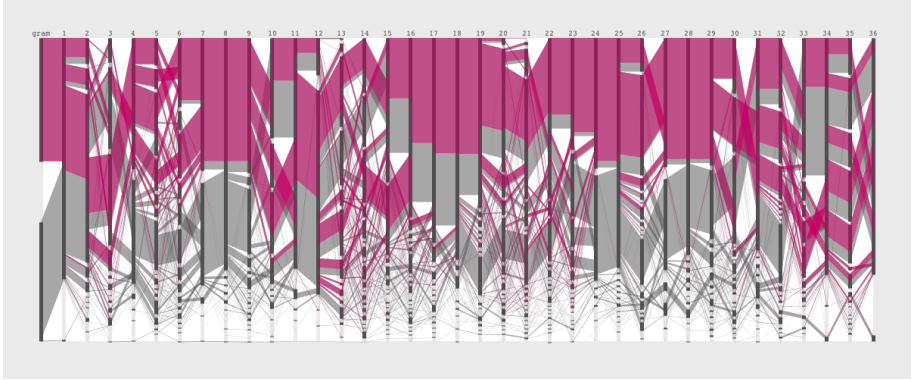

Figure 4.8: A data sketch with the parallel set vis idiom to compare the proportion of two alignment sets.

following data sketch is a combination of previous versions. The goal was to improve the visual search for the difference in sequential conservation between two sets (Figure 4.9). In this visual encoding, the height of bars is scaled to a logarithmic scale to condense the height, and thin transparent lines were used to represent individual amino acid sequences. The lines are positioned so that they are rendered as bundles. The amino acids at each position are ordered to emphasise the difference between two sets. For instance, those amino acid residues that are common in both sets are placed close to the midline. If a residue at a position is unique or predominant for one set, they are placed away from the midline.

This iteration is an example of trying different heuristics, such as sorting items, to improve the visual search for target analysis tasks. Sometimes the choice of heuristics determines whether you see or miss a relevant unexpected pattern. For example, a bundle of blue lines from position 19, following A, K, A, D, V, and C, is a subgroup unique to the gram-positive set. This subgroup is, in fact, identifiable in the previous iterations, but it is more readily perceived in this data sketch.

This iterative data sketching process has led to the final design of the Sequence Diversity Diagram (SeDD) (Figure 4.10). The key modification from the previous data sketch is the use of vertical position to encode the identity of residue and to avoid the need to label each amino acid residues. The line thickness encodes the frequency. The colour categorises two sets. The horizontal gaps separate the functional groups of amino acids, and the less frequent sequential patterns are filtered out to improve the salience to the strong conservation signals. The choice of colour (cyan and magenta) is intentional: the gram-positive is colour-coded

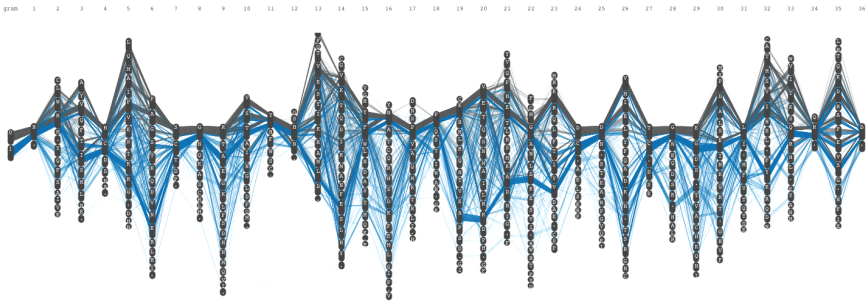

Figure 4.9: A data sketch to encode bundles of sequence alignments. The height of residues was scaled logarithmically, and two sets were distinguished by colour.

in cyan, the gram-negative is colour-coded in magenta, and the overlapping portions are coloured in purple. This categorical colour scheme improved visual scanning for positions that are similar and different between two sets. The preliminary evaluation of SeDD is discussed in Section 4.3.

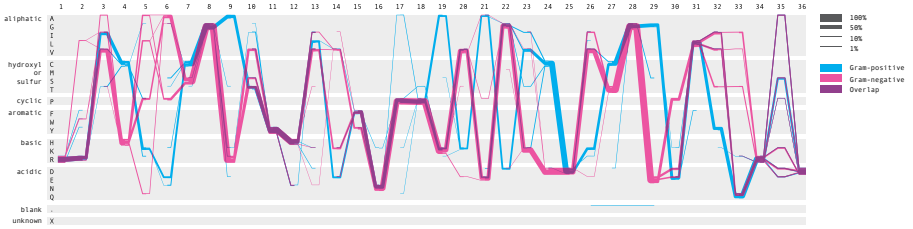

Figure 4.10: The final iteration of data sketch. This visual encoding is named as the Sequence Diversity Diagram.

In summary, the data sketching has three key concepts: 1) the use of real data sets, 2) the iterative design process, and 3) the speed over polish. The intermediate steps in a design process are often not discussed in published papers, but the small incremental design improvements are essential and informative aspects of a design study. This personal motivation led to include these descriptive details in a hope that it is useful for other vis designers.

Data sketching entails generating visual encoding ideas and editing them to match user tasks iteratively. Sketches in general serve to amplify a designer's creative thinking and relieve limited cognitive capacity of working memory [77]. As shown in Figure 4.11, one sketch led to another idea, and sometimes

ideas were merged to form a new idea. Steven Johnson elaborates on Stuart Kauffman's idea of the *adjacent possible* in his book [78]. The concept of the *adjacent possible* is that the world is capable of extraordinary change at any moment, but only certain changes can happen. Johnson argues that good ideas are not conjured out of thin air; they are a result of a continual exploration of the adjacent possible.

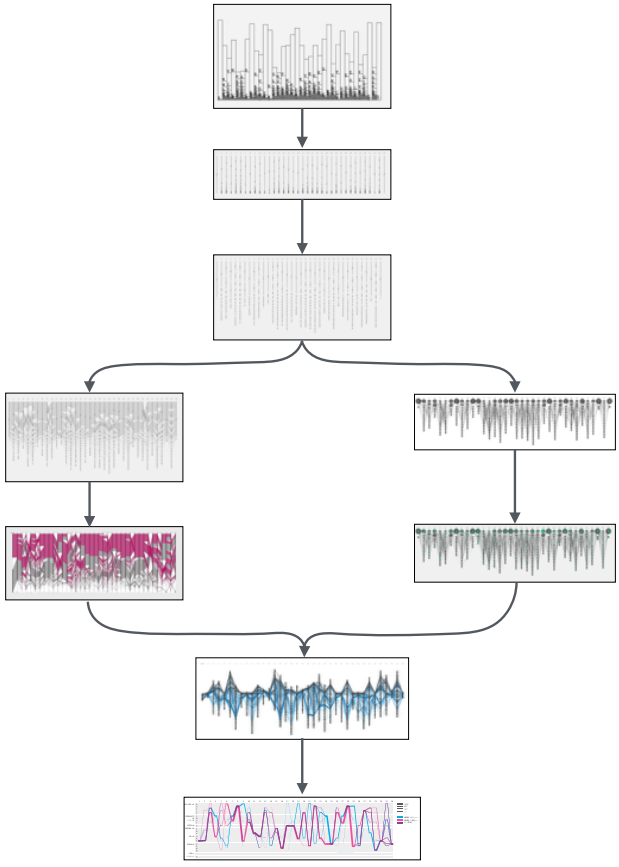

Figure 4.11: The iterative design process of Sequence Diversity Diagram.

In the context of the visualization design process, there are three types of ideas: ideas that are realised, ideas that have not been realised (adjacent possible) and ideas that are unknown or unattainable (see Figure 4.12). For example, *Idea a* is a starting point of the process. The first prototype you realise is *Idea b*.

Only when you realize *Idea b*, *Idea c and d* become potential subsequent designs that extend on *Idea b*. As long as *Idea c or d* are not realised, *Idea e, f and g* are likely to remain as unknown and unattainable. Thus, we advocate data sketching to realise the *adjacent possible* and to explore the vis design space.

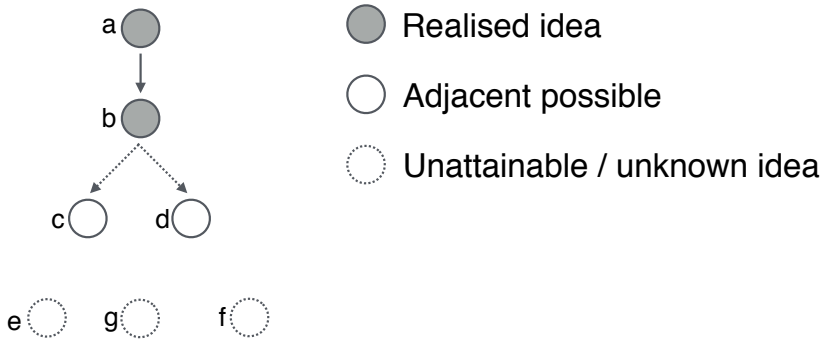

Figure 4.12: The network of ideas in the vis design space. There are three types of ideas: ideas that are realised (a and b), ideas that have not been realised (c and d), and ideas that are unknown and unattainable (e, g, and f).

4.3 Sequence Diversity Diagram for Comparative Analysis of Multiple Sequence Alignments

R. Sakai and J. Aerts, "Sequence Diversity Diagram for comparative analysis of multiple sequence alignments," *BMC Proc.*, vol. 8, no. Suppl 2, p. S9, 2014.
R. Sakai and J. Aerts, "Erratum to : Sequence Diversity Diagram for comparative analysis of multiple sequence alignments," *BMC Proc.*, vol. 8, no. Suppl 2, p. S10, 2014.

4.3.1 Abstract

Background: The sequence logo is a graphical representation of a set of aligned sequences, commonly used to depict conservation of amino acid or nucleotide sequences. Although it effectively communicates the amount of information present at every position, this visual representation falls short when the domain task is to compare between two or more sets of aligned sequences. We present a new visual presentation called a Sequence Diversity Diagram and validate our design choices with a case study.

Methods: Our software was developed using the open-source program called Processing. It loads multiple sequence alignment FASTA files and a configuration file, which can be modified as needed to change the visualization.

Results: The redesigned figure improves on the visual comparison of two or more sets, and it additionally encodes information on sequential position conservation. In our case study of the adenylate kinase lid domain, the Sequence Diversity Diagram reveals unexpected patterns and new insights, for example the identification of subgroups within the protein subfamily. Our future work will integrate this visual encoding into interactive visualization tools to support higher level data exploration tasks.

4.3.2 Background

The sequence logo [74] has been the most adopted graphical representation for the multiple sequence alignments of nucleotides or amino acids. Its popularity among biologists stems from its simplicity and accuracy in visually communicating the motif or signature of aligned sequence by contrasting the conserved and diverse positions by the height of single letter codes. It emphasizes the most conserved positions effectively, but this visual encoding falls short for comparative analysis of multiple groups of aligned sequences.

In order to compare two sets of multiple sequence alignments, for instance the adenylate kinase lid (AKL) domain of Gram-negative bacteria and Gram-positive bacteria, it requires three figures for analysis: a sequence logo for across all organisms and one for each subfamily. Although each sequence logo efficiently represents and summarizes the amount of information present at every position, it requires high cognitive load to scan back and forth between figures to identify key positions that are shared or differ between two subfamilies. In addition, the sequential frequency of consecutive positions is missing in the traditional sequence logo.

In this paper, we introduce a new visual for presenting multiple sequence alignments for comparative analysis to aid the domain expert in comparing two or more sets of sequence alignments. We set three objectives for the redesign: improving the visual comparison of two sets of aligned sequences, encoding the sequential conservation, and reducing the visual noise. We discuss the strengths and weakness of the sequence logo for the comparative aligned sequence analysis. We also elaborate on our choices for visual encoding and validate our design choices with a case study of the AKL domain dataset, provided through the BioVis 2013 redesign contest. We conclude with our future work to develop an interactive visualization tool for explorative data analysis of multiple sequence alignments.

Design rationale

A sequence logo figure consists of a stack of letters, representing an amino acid residue or a nucleotide with varying heights to encode the information content of the residue at every position. The information content, also known as the Shannon Entropy [75], is a measure of the uncertainty in a random variable. In the case of sequence logos, this uncertainty is interpreted as how well residues are conserved at each position. For the amino acid sequence alignment, the residues are typically grouped by their structure and the chemical characteristics of their R groups and each residue is colour-coded according to the assigned group. In

order to compare two groups of aligned sequences, three separate figures are generated: one that combines both samples and one for each group. Although the sequence logo representation may be effective for individual subfamilies, it is not designed for comparing multiple alignment sets.

Another motivation for improving on the traditional sequence logo is to encode the sequential information of conserved positions. For example, the Proline (P) and the Lysine (L) at position 5 and Alanine (A), and Proline (P) at position 6 are the most conserved amino acid residues [Figure 4.13], but the fact that the Proline (P) is followed by the Alanine (A) and the Lysine (L) is followed by the Proline (P) is not clear from this figure. Thus, conveying and communicating the sequential conservation of adjacent and distant residues may aid domain experts to gain new insights or reveal unexpected patterns as it helps to analyze the sequence alignment as sequence, rather than individually by position.

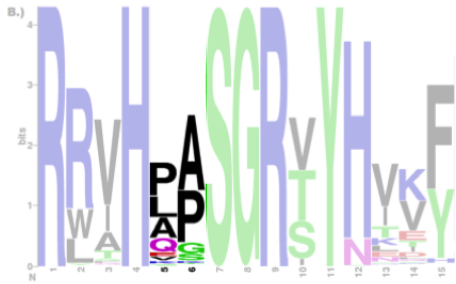

Figure 4.13: Sequence logo of the adenylate kinase lid domain from Gram-negative bacteria. It is not clear whether the Proline (P) at position 5 is followed by the Alanine (A) or the Lysine (K) at subsequent position 6. Only positions from 1 through 15 are shown and masked except for position 5 and 6.

Related works

The new work was inspired by the Parallel Sets (ParSet) [76] and the ProfileGrids [79]. ParSet is an interactive visualization application for multi-dimensional categorical data. For each dimension, each of its categories is connected to a number of categories in the subsequence dimension, showing the subdivision of categories. When this visual encoding is applied to the multiple sequence alignment data [Figure 4.14], it encodes the sequential frequencies, but the functional grouping of amino acids is lost and each vertical bar, representing a frequency at the position, needs to be labeled to indicate which amino acid it is.

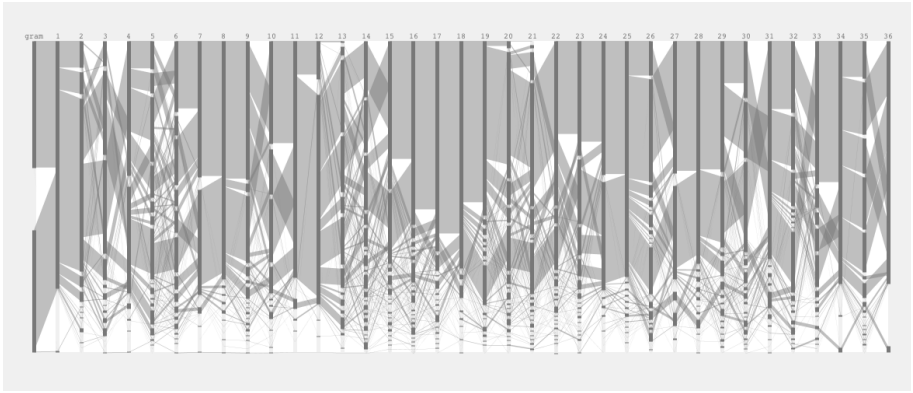

Figure 4.14: Parallel Sets representation of the adenylate kinase lid domain. Both Gram-positive and Gram-negative sequence positions are laid horizontally on the x-axis. Two sets of sequence alignment are separated as categorical data, as shown in the first dimension.

ProfileGrids use a visualization which reduces alignments to a matrix, colored according to the residue frequency at each position. Although this visual encoding provides a concise overview of alignments, it does not encode the sequential frequency and still requires multiple matrixes for the comparative analysis.

Sankey diagrams [80] are a type of flow diagram, commonly used to visualize energy or material flows between processes within a system, where the thickness of the arrows is proportional to the quantity. One advantage of using the Sankey diagram over the Parallel Set is that the line or arrow thickness is consistent between dimensions, thus minimizing the effect of the line width illusion [81].

Jalview is an interactive visualization tool for editing, analysis, and annotation of multiple sequence alignments [82]. It is a comprehensive tool which can work with sequence annotation, secondary structure information, phylogenetic trees and three-dimensional molecular structures. Its multiple alignment views allow the same data to be viewed independently in many different ways at the same time, but it requires more than one panel to compare multiple samples.

4.3.3 Methods

Implementation

An open-source, java application developed in Processing [70] is available for Linux, Mac OS and Windows. It loads FASTA files of multiple sequence alignments and a configuration file with visualization parameters, such as grouping and coloring schemes. The aligned set of not only protein sequences, but also nucleotides can be visualized by modifying the configuration file. The application also allows interactive edits to the image and can export PDF or PNG images. The application, source code, and wiki are available at https://bitbucket.org/biovizleuven/sequencediversitydiagram.

4.3.4 Results

Visual encoding

We introduce a new visual encoding, called a Sequence Diversity Diagram, for comparative analysis of multiple sequence alignments [Figure 4.15]. The amino acid positions are plotted horizontally on the x-axis. Amino acid residues and their groupings are plotted on the y-axis. The grouping of amino acids based on their structure and the general chemical characteristics is visually enhanced with horizontal gaps. On this grid layout, a flow diagram is drawn to represent sequence alignment frequency and each sample is color-coded. The line weight represents the relative frequency of residues at the two consecutive positions with respect to the total number of sequences. By linking two adjacent positions, it visualizes the sequential frequency and co-occurrence between two adjacent positions. By overlaying two samples, both the conserved and diverse positions are studied in a single figure.

Case study

The Sequence Diversity Diagram [Figure 4.15] is generated from the multiple sequence alignments of the adenylate kinase lid (AKL) domain of Gram-positive and Gramnegative bacteria, provided by the BioVis 2013 contest. By overlaying two sets of sequence alignments and color-coding each, a figure representing both multiple sequence alignments is generated. The positions conserved in both samples are shown in purple [for instance, amino acid 11] and positions that differentiate two subfamilies are found where the lines are not overlapping [amino acid 24]. Less frequent pairs of positions, less than one percent of the

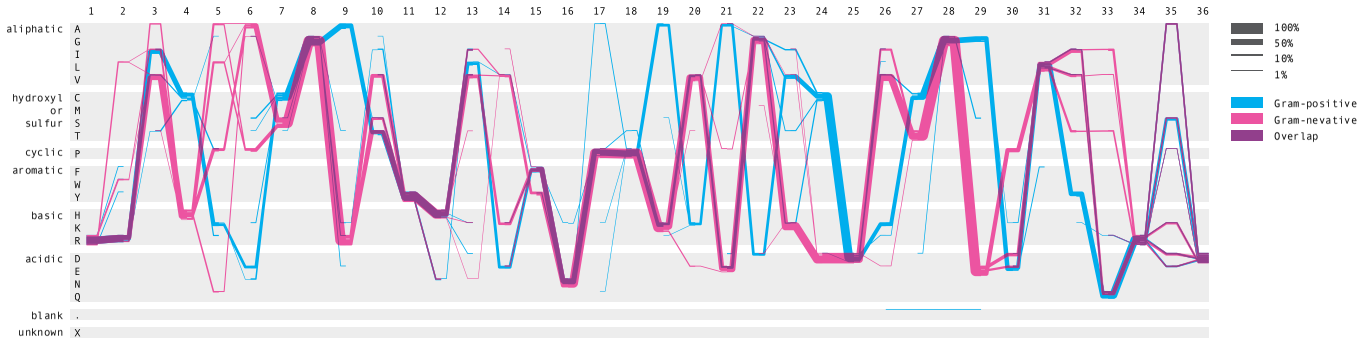

Figure 4.15: Sequence diversity diagram of the adenylate kinase lid domain. The sequence positions are laid horizontally on the x-axis and the residue groupings are shown on the y-axis. The Sequence Diversity Diagrams represent two samples, color-coded and overlaid on top of each other. The line weight represents the relative frequency of residues at the two consecutive positions with respect to the total number of sequences.

total number of aligned sequence in this figure, are filtered out to improve the readability and to reduce the visual noise via the user interface interaction.

The redesigned figure encodes the sequential frequency, and it becomes clear that the Proline at position 5 is followed by the Alanine at position 6. Encoding the sequential frequency also helps to identify subgroups, and another example is a subgroup of Gram-positive from amino acid 19. This subgroup may appear to be "out of phase" oscillations due to a single amino acid insertion or deletion. However, it is unlikely that this is an phasing error, because these subgroups do not share an exact repeated motif even with a phase offset, and the input data are curated structural alignments.

4.3.5 Conclusions

The Sequence Diversity Diagram is designed to improve the task of visual comparison between multiple sets of aligned sequences, such as protein subfamilies. This Sankey-like diagram is plotted on a grid layout of positions and amino acid functional groups. Instead of having separate sequence logo figures, it consolidates multiple sets of sequence alignments into a single figure. It also encodes the sequential conservation, which may lead to new insights or unexpected findings, such as the identification of the subgroup in the Gram-positive bacteria.

The case study with the AKL domain data demonstrates that this visual encoding is useful for comparative analysis and the result has encouraged us to develop interactive features to further support data exploration tasks. Our future work includes calculation of the mutual information to detect the dependency of one position on another [83]. Then, linking these co-evolving position pairs to the physical proximity or the structural conformation in 3D by loading a protein data bank (PDB) file in Jmol [84]. The aim is to support explorative data analysis at the level of sequence alignment, information theory, and three-dimensional structure. Lastly, we plan to develop a plug-in for widely used applications for multiple sequence alignment analysis, such as JalView, and a reusable BioJS component [85].

Acknowledgements

We gratefully acknowledge the dataset provided by Drs. Magliery and Sullivan at The Ohio State University for the purposes of the BioVis 2013 Contest. Research supported by: KU Leuven program financing PFV/10/016 SymBioSys, IWT 020 ExaScience Life Pharma, iMinds Art&D Instance, and COST: Action BM1104.

This article has been published as part of BMC Proceedings Volume 8 Supplement 2, 2014: Proceedings of the 3rd Annual Symposium on Biological Data Visualization: Data Analysis and Redesign Contests. The full contents of the supplement are available online at
http://www.biomedcentral.com/bmcproc/supplements/8/S2

Chapter 5

Sequential Tasks

Section 5.3 is reprinted from the following publication:

C. W. Bartlett, S. Y. Cheong, L. Hou, J. Paquette, P. Y. Lum, G. Jäger, F. Battke, C. Vehlow, J. Heinrich, K. Nieselt, R. Sakai, J. Aerts, and W. C. Ray, "An eQTL biological data visualization challenge and approaches from the visualization community," *BMC Bioinformatics*, vol. 13 Suppl 8, no. Suppl 8, p. S8, Jan. 2012.

The author's entry was selected for the Biology Experts Pick award for the BioVis data contest 2011. The author contributed the subsection describing the entry.

Figure 5.8 is reprinted from the following publication:

K. Pougach, A. Voet, F. a. Kondrashov, K. Voordeckers, J. F. Christiaens, B. Baying, V. Benes, R. Sakai, J. Aerts, B. Zhu, P. Van Dijck, and K. J. Verstrepen, "Duplication of a promiscuous transcription factor drives the emergence of a new regulatory network," *Nat. Commun.*, vol. 5, p. 4868, 2014.

Understanding analysis tasks is a basic requirement for designing an effective visualization system. A task may be a single visual query. But, in many cases, an analysis process involves a sequence of tasks, where findings in one task feed into the subsequent task. In this chapter, two design studies are reviewed: a visual analytics tool for expression quantitative trait loci (eQTL) data analysis (Aracari) and a visualization tool for comparative analysis of multiple sequence alignments of amino acid sequences (Seagull). Even though Aracari and Seagull encompassed different biological domains and datasets, both tools were designed to support a sequence of analysis tasks by linking multiple views of the data. Individual tasks, as well as the flow of one task to another, are an important aspect of vis design.

5.1 Case Study: Aracari

Variations in deoxyribonucleic acid (DNA) sequence between individuals can generate variations in the amount of ribonucleic acid (RNA) produced, and therefore the amount of protein, potentially creating differences in function, fitness or health. Mapping the DNA variants that have these functional effects on gene expression is called eQTL analysis. While the technology to acquire data to detect these effects has only recently become available, it is expected that networks of directly or indirectly interacting DNA polymorphisms non-linearly affect specific gene expression levels, which makes mapping these DNA variations a particular challenge. Additionally, eQTL signals are dependent on both tissue and developmental time point. Besides these challenges, understanding eQTL signals is critical to understanding how DNA variation creates individual responses to the environment - differences that are not necessarily identical across individuals, or even different tissues in the same individual. This is important since it makes surrogate tissue analysis problematic.

The Symposium on Biological Data Visualization (BioVis) hosted data contests on eQTL data in 2011 and 2012. Although the BioVis contest 2012 was canceled eventually, we prepared an entry and developed an interactive visualization tool, called Aracari. This vis tool examined the association patterns of genome sequence variations and expression levels. The analysis tasks were to provide visualizations to identify the pattern of genome sequence variations and expression levels, that predict the occurrence of a hypothetical disease. More specifically, the contest challenge included the characterisation of both direct and indirect single nucleotide polymorphism (SNP) effects on the genes (cis or trans effect) and gene-gene interactions.

Aracari consists of two analysis modes: the gene expression view and the SNP

view. The gene expression view was designed to compare the distributions of gene expression levels between the unaffected and affected individuals (Figure 5.1). This view provides three different visual encoding options to compare distributions: histograms, Q-Q plots, and locally weighted polynomial regression curves (Figure 5.2). After examining the distributions, the user could select and note a list of genes. The list is displayed on the right side of the interface, which remained consistent between two analysis modes. The SNP view was designed to examine single and two-locus statistical analysis data [86], provided as a part of the contest data (Figure 5.3). The task in this mode was to identify a set of SNPs that affect the gene expression directly or indirectly. The genes from the preceding view were used to guide identification of candidate SNPs. Similarly, the SNPs of interests could be added to the list on the right panel. The record of the analytical reasoning process is referred to as *analytic provenance* [87]. Given the SNPs of interest, the user could filter samples based on genotypes in the gene expression view to link the genotype information and the distributions of affected and unaffected samples.

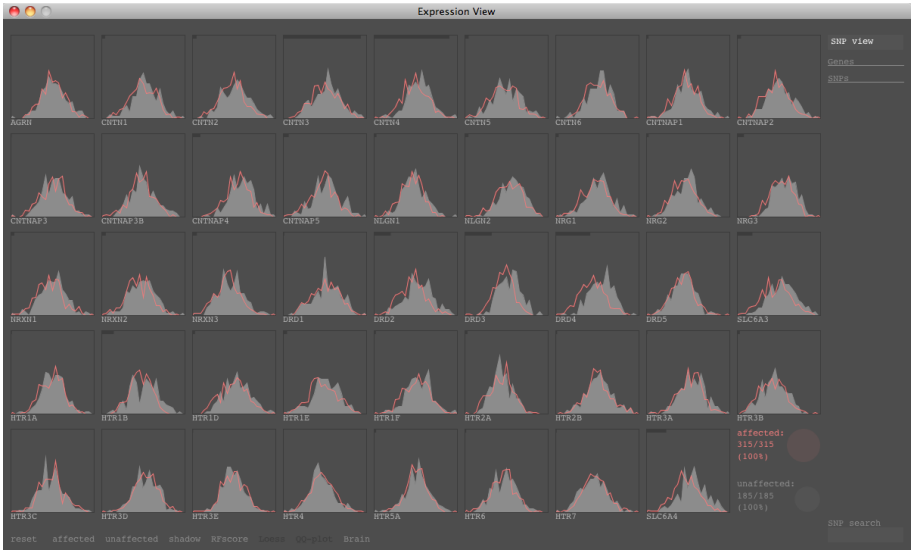

Figure 5.1: The gene expression view. Small multiples of differential histograms examine how the expression levels are distributed differently between affected and unaffected.

In this design study, a vis tool was developed to provide the context for interpretation of the outputs of the integrative computational methods, namely single and two-locus PLINK results. Aracari provides two viewing modes to analyse the gene expression and the PLINK analysis results. With advances in

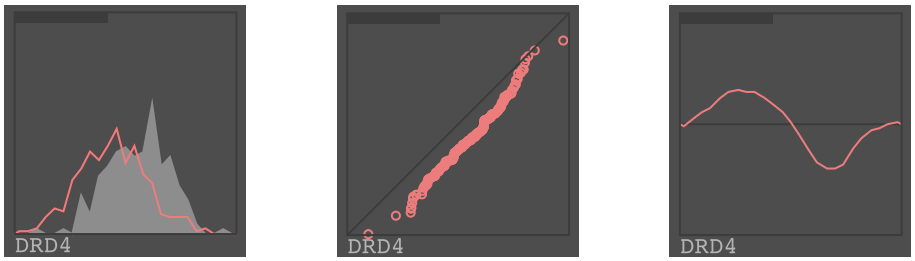

Figure 5.2: Three representations comparing the distributions of affected and unaffected individuals, based on the *DRD4* gene expression level of the brain tissue.

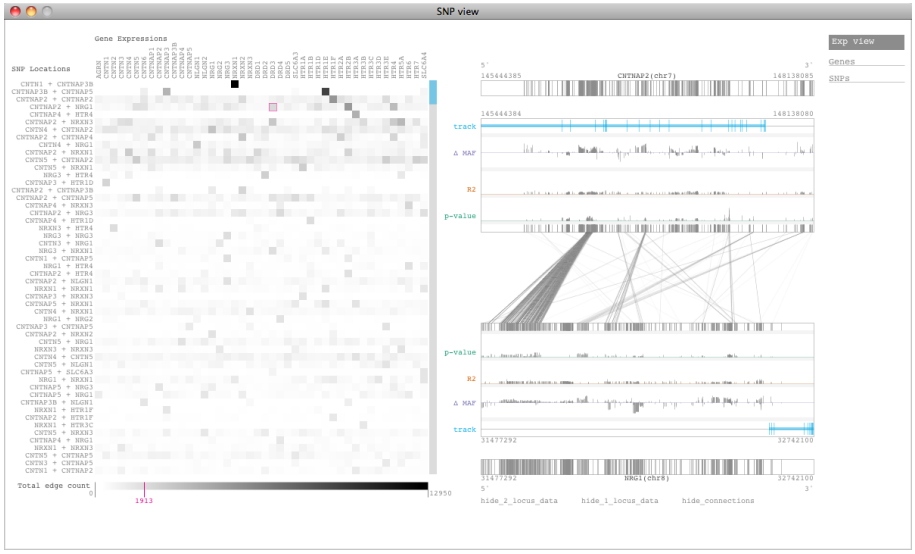

Figure 5.3: The SNP view, visualising the single and two-locus analysis results.

integrative approaches for high-throughput experiments' data in biology, the visualization challenge is not only to show the outputs, but also to provide the context to understand and evaluate the result by including the input datasets used for the computational method. The development process of such vis tools requires analysing the intricacy of tasks and analytical reasoning process to inform the choice of visual encoding and interaction design. If a tool involves multiple viewing modes, it is also important to carefully consider the continuity of tasks, as well as the consistency in the layout.

5.2 Case Study: Seagull

Collaboration:

Justin Ashworth[1]

J.A. implemented the mutual information calculation code and discussed the design of the Seagull prototypes with the candidate.

[1]*Institute for Systems Biology, Seattle, WA, USA*

Seagull is an interactive visualization tool designed for comparative analysis of multiple sequence alignments (MSAs). It includes three interactive visual representations to study sequence alignments, variation and co-variation of positions, and the structural information. The main motivation for developing Seagull was to support analysis by linking the sequence alignment information to the structural information. The key concept in proteomics is "structure and function"; The structure of a protein determines its function. Analysis of MSAs typically relies on positional statistics, as seen in sequence logos, to infer the conservation of residues in a linear representation of amino acid sequences. However, these amino acid sequences are in fact folded into 3D structures, and the structure is critical for understanding its function. Thus, the goal is to make an exploratory visual analytics tool by supporting transitions of analysis tasks from the linear representation to the 3D structure of the protein. For the demonstration of the tool, the MSAs of the adenylate kinase lid (AKL) domain from gram-negative and gram-positive bacteria provided by BioVis 2013 are used.

The first representation is a Sequence Diversity Diagram (SeDD), visualizing the sequential conservation to improve the comparison of multiple sets of MSAs (Figure 5.4). The design process of SeDD is discussed in Chapter 4. The second view is a visualization of both consensus and correlation values of each pair of positions in a circos-like layout to identify positional dependencies (Figure 5.5)[58]. Covariance of two distant positions in its amino acid sequence implies their proximity in 3D structure and its importance for the protein structure, function and dynamics [83]. The last view is a Java molecule viewer (Jmol) [84]. This Jmol view is linked to the two previous views to examine the physical proximity of selected positions in 3D based on a Protein Data Bank (PDB) file (Figure 5.6). Seagull supports data exploration analysis by bridging three different levels of information: sequence alignments, sequence statistics, and the structural information.

By combining vis components designed for specific tasks, the experts were

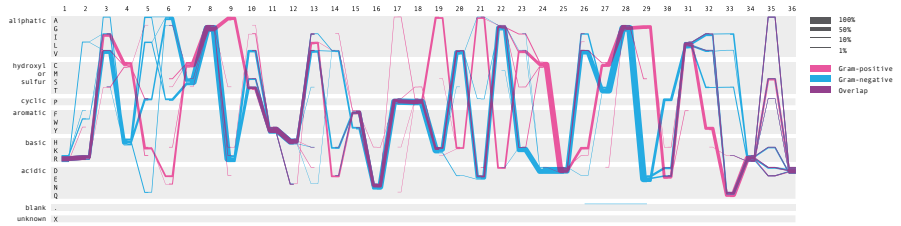

Figure 5.4: Sequence Diversity Diagram for comparative analysis of multiple sets of multiple sequence alignments of the adenylate kinase lid domain.

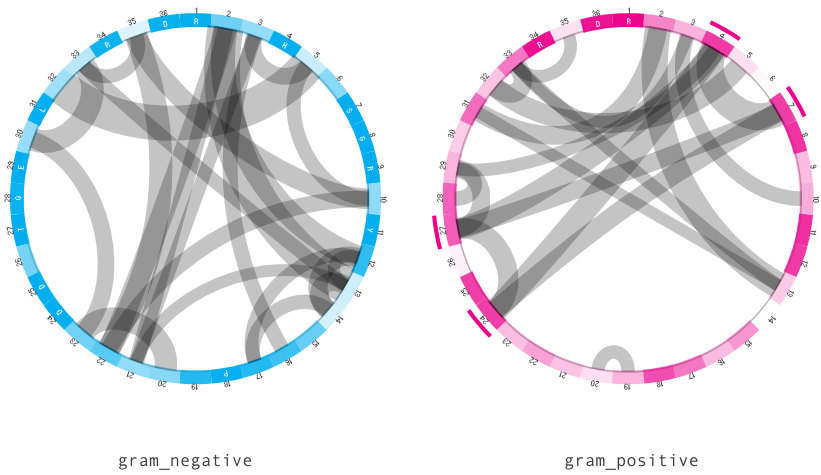

Figure 5.5: Visualization of information content per position and mutual information between positions. In the gram-positive sample, position 4, 7, 24 and 27 are selected. These positions have relatively high conservation scores as encoded in saturation, and the mutual information between positions are represented in the thickness of grey lines inside the circle.

able to identify a residue or a set of residues of interest and to study their position and orientations in the 3D folded protein structure. Seagull integrated both novel vis idioms and an existing vis tool. The circular representation of information content and mutual information was preferred over the conventional heatmap visualization of adjacency matrices, especially for cases where the difference between MSA sets are subtle. Figure 5.7 shows a comparison of two vis idioms for mutual information, using the sequence alignments of the

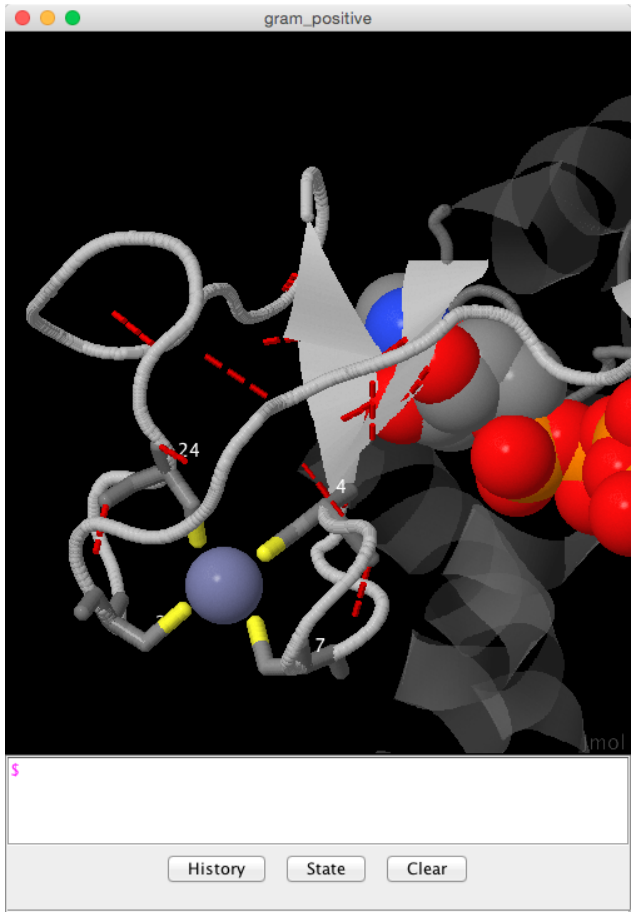

Figure 5.6: The view of selected amino acids in Jmol. In the gram-positive sample, the structure is stabilised by a bound metal ion, tetrahedrally coordinated by the Cysteine at position 4, 7, 24 and 27.

Krueppel-type ZNF. The subtle difference in mutual information score among species is difficult to spot in the heatmap representation, while you can more readily identify those differences in the circular layout. The Seagull vis idiom was also applied to nucleotide sequence alignments to study DNA-binding sites [88] (Figure 5.8).

The design studies of Aracari and Seagull demonstrate that analysis tasks in biology are often not a single question but rather consist of a sequence of linked questions and tasks. A vis design strategy is to decompose a sequence of tasks

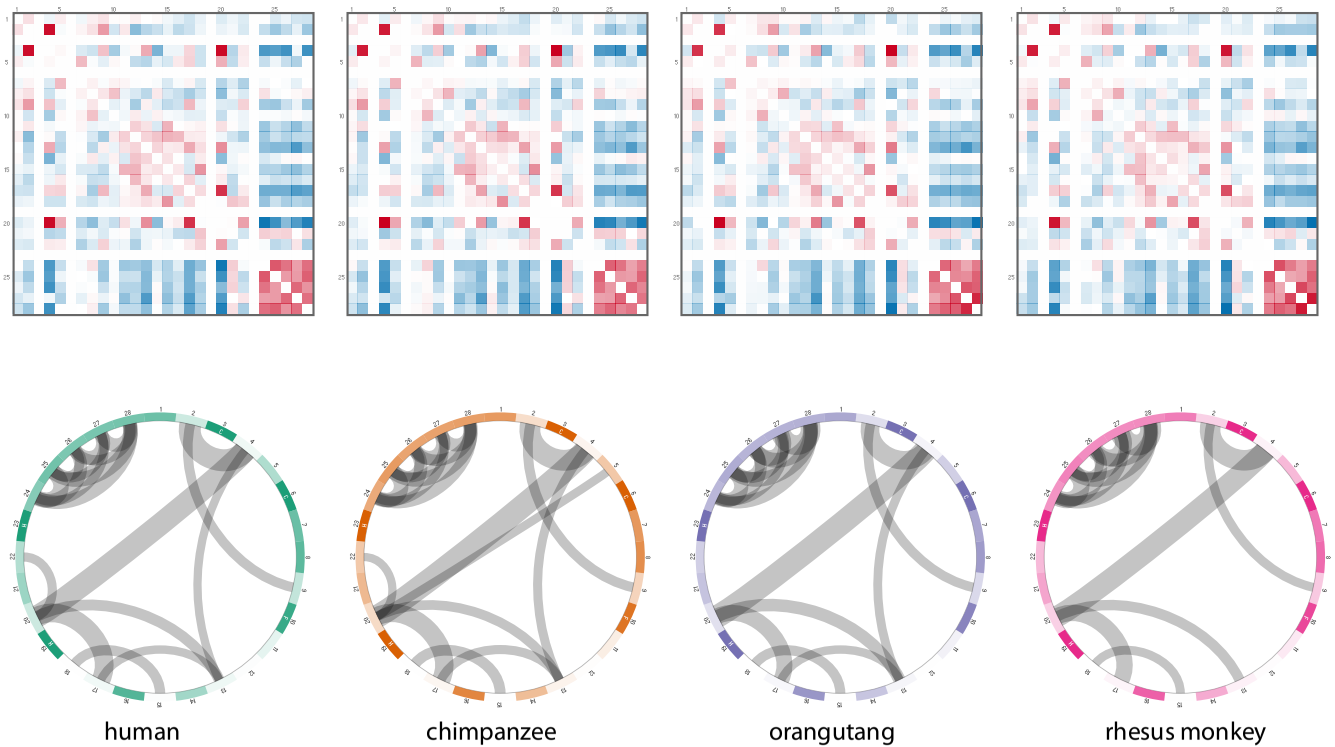

Figure 5.7: Comparison of vis idioms for visualizing the mutual information. The Krueppel-type ZNF sequence alignments of human, chimpanzee, orangutang and rhesus monkey are compared. (MSA data provided by Prof. Katja Nowick)

into individual tasks, then to consider and choose the vis idiom for each task, and provide signifiers to link each visual component [57].

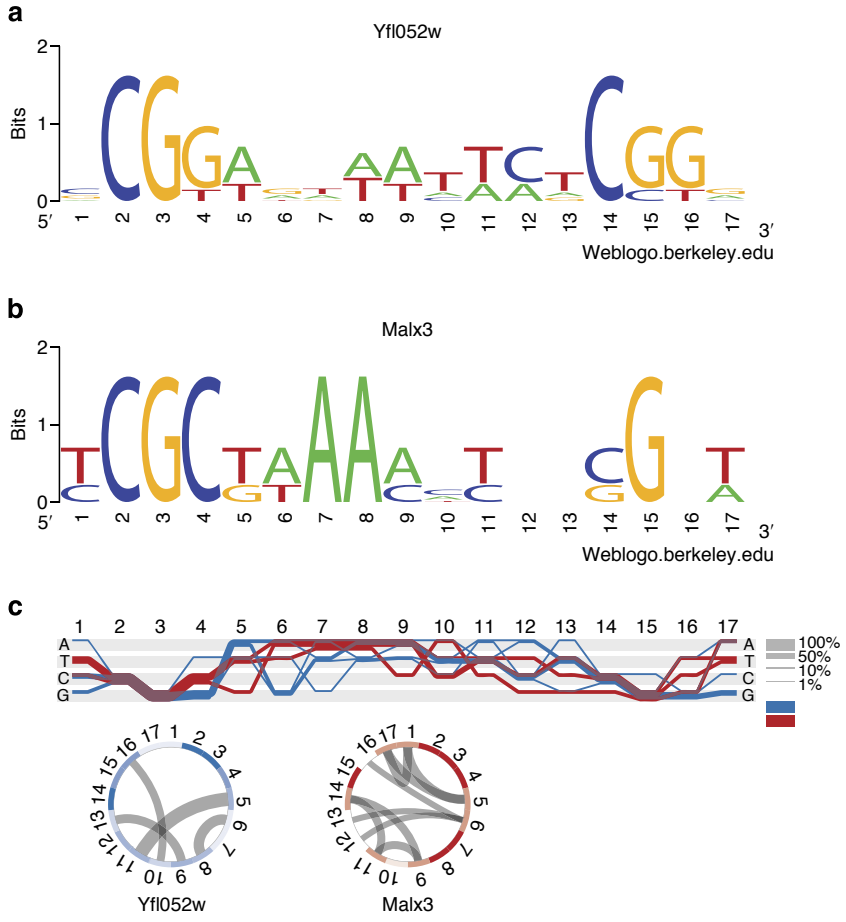

Figure 5.8: Different DNA-binding specificity of different MalR transcription factors. (a) Sequence logo of Yfl052w DNA-binding site CGG(9N)CGG. (b) Sequence logo of Malx3 DNA-binding site CGC(9N)CGN. (c) Sequence Diversity Diagram conveying differences and similarities between Yfl052w (blue) and Malx3 (red) binding sites. Regions where two groups overlap are shown in purple. Thickness of the line shows the relative proportion of the aligned sequences. Positions that differentiate two groups are identified by a separation of blue and red lines. The rings below are basic heatmaps in circular layout showing the information content of the positions. The more saturated the colour, the higher the information content, thus indicating conserved regions. Mutual information (MI) represents covariance of positions and is shown as grey lines inside the circles. High MI indicates the dependency of one position on another. (Figure 2 in [88])

5.3 An eQTL Biological Data Visualization Challenge and Approaches from the Visualization Community

Section 5.3 presents excerpts selected from the publication shown below to give a brief introduction to the data contest, and to present the judge's comment as well as comparison of awarded entries. Our entry was selected for the Biology Experts Pick award. Our contribution to the paper is under the heading of **Biology experts' pick: Ryo Sakai and Jan Aerts**.

C. W. Bartlett, S. Y. Cheong, L. Hou, J. Paquette, P. Y. Lum, G. Jäger, F. Battke, C. Vehlow, J. Heinrich, K. Nieselt, R. Sakai, J. Aerts, and W. C. Ray, "An eQTL biological data visualization challenge and approaches from the visualization community.," *BMC Bioinformatics*, vol. 13 Suppl 8, no. Suppl 8, p. S8, Jan. 2012.

5.3.1 Abstract

In 2011, the IEEE VisWeek conferences inaugurated a symposium on Biological Data Visualization. Like other domain-oriented Vis symposia, this symposium's purpose was to explore the unique characteristics and requirements of visualization within the domain, and to enhance both the Visualization and Bio/Life-Sciences communities by pushing Biological data sets and domain understanding into the Visualization community, and well-informed Visualization solutions back to the Biological community. Amongst several other activities, the BioVis symposium created a data analysis and visualization contest. Unlike many contests in other venues, where the purpose is primarily to allow entrants to demonstrate tour-de-force programming skills on sample problems with known solutions, the BioVis contest was intended to whet the participants' appetites for a tremendously challenging biological domain, and simultaneously produce viable tools for a biological grand challenge domain with no extant solutions. For this purpose expression Quantitative Trait Locus (eQTL) data analysis was selected. In the BioVis 2011 contest, we provided contestants with a synthetic eQTL data set containing real biological variation, as well as a spiked-in gene expression interaction network influenced by single nucleotide polymorphism (SNP) DNA variation and a hypothetical disease model. Contestants were asked to elucidate the pattern of SNPs and interactions that predicted an individual's disease state. 9 teams competed in the contest using a mixture of methods, some analytical and others through visual exploratory methods. Independent panels of visualization and biological experts

judged entries. Awards were given for each panel's favorite entry, and an overall best entry agreed upon by both panels. Three special mention awards were given for particularly innovative and useful aspects of those entries. And further recognition was given to entries that correctly answered a bonus question about how a proposed "gene therapy" change to a SNP might change an individual's disease status, which served as a calibration for each approaches' applicability to a typical domain question. In the future, BioVis will continue the data analysis and visualization contest, maintaining the philosophy of providing new challenging questions in open-ended and dramatically underserved Bio/Life Sciences domains.

Visualization and analytical complexity

eQTL analysis provides a target-rich domain for visualization and visual analytics approaches. With the goal of "convey how it works", across data with potentially millions of variables, just the sheer size makes visual abstraction and summarization a practical necessity. The complex and conditional interrelations, and the necessity of communicating these as a goal, further cements the importance of visualization to this domain. While one might think of an eQTL data set as being represented by a graph with nodes representing genomic loci, and edges representing relationships, the requirements for eQTL analysis and representation go beyond traditional network/graph representation techniques, and no extant technique is completely adequate to convey the conditional, and biologically error-laden results.

Even raw statistical analysis of this data is problematic. It is fairly easy to analyze single-locus direct effects where, all other things being equal, the presence of a particular allele at some locus predisposes an expression level to be elevated or depressed. This can be easily accomplished with the popular analysis program PLINK [86]. It is harder to analyze multi-locus direct effects, where the specific alleles at a pair of loci modulates expression. It becomes computationally intractable to calculate indirect effects where a complex combination of an unknown number of alleles interact in affecting an expression level, or combination of expression levels. And of course, even if the raw statistics could be calculated, thousands or millions of ranked lists of millions of interacting SNPs and expression levels, with each list potentially depending on numerous factors, would be impossible to interpret directly.

Using the array of commonly available tools (summarized here [89]), only small slices of the eQTL visualization problem can be effectively tackled. The utility of such a piecewise approach is highly dependent upon the judgment and skill of the user, and the best way to approach this data and its analysis, is as yet

undefined. Static or animated, fixed representation or interactive, exploratory or explanatory, displaying statistics, or guiding calculations to perform, it is hard to imagine any representation that cannot provide some useful insights into the data, and equally hard to imagine any that come close to being completely adequate for all uses. In the 2011 BioVis contest, entrants explored a large range of themes, and demonstrated tools that applied several of these themes.

Spiked-in network

The spiked-in network (Figure 5.9) was modeled as a series of correlations in a 15 $\times$ 15 matrix to express the gene $\times$ gene interaction, then an additional dimension was added in to allow for specific effects of the 3 possible genotypes at single SNP in each gene, where this single SNP was the only genetic variant in the gene that affects gene expression in the network (as described in Data processing section). The resulting correlation matrix, which due to our standardization procedures could be called a variance-covariance matrix, is not ideal for further statistical analysis since it not a properly formulated, symmetric positive definite matrix. Therefore the closest proper variance-covariance matrix was estimated [90] and used for the simulation. Using the R statistical language framework [91], the mvtnorm [92, 93] library function "rmvnorm" was used to simulate random multivariate normal data using singular value decomposition on this variance-covariance matrix and genotypic means estimated in the data processing step (above). This simulation was conducted for each simulated person in the dataset conditional on the genotypes from the data shuffling step. The result is 15 gene expression values for each of 1000 simulated persons. The gene expression values were finally spiked-in by convolving the gene expression values from data shuffling with the spiked-in network multiplied by a weighting parameter. The weight of the spiked-in data was varied for each set of simulations where the spiked-in network was up-weighted in the first practice dataset (to make the network easy to find) and reduced on each consecutive iteration of practice datasets with the official contest data having the smallest value, and therefore these effects were harder to detect in the contest versus practice.

Sakai and Aerts

This entry provided two exploratory tools, one to investigate the effect of gene expression on the disease, and one to investigate the effect of SNP genotype on gene expression. The expression-disease tool provided an interactive interface using (modified) parallel coordinates [94], which presented all of the individuals and expression levels simultaneously, and enabled the user to identify relevant factors through a visual analytics paradigm. Simple differential histograms for each gene expression in affected and unaffected individuals, and coloring of each individual's trace based on affected or unaffected status, provided an interface

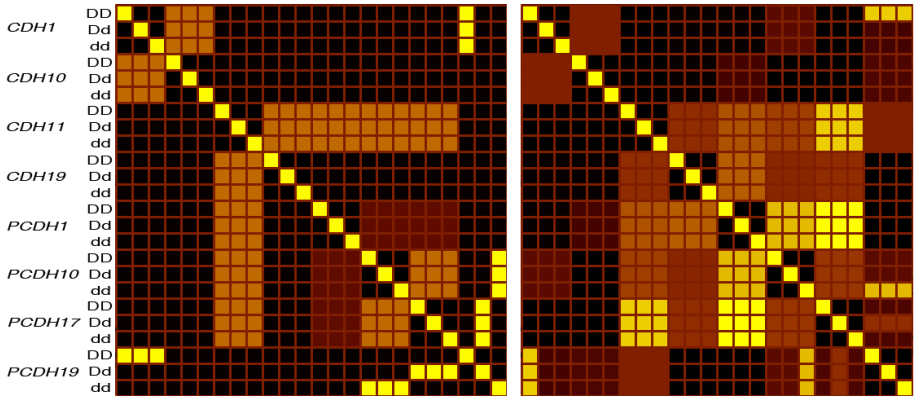

Figure 5.9: A heat map representation of the spiked-in correlation network in the simulated data. The heatmap is a two dimensional projection of a four dimensional matrix, 15 × 15 genes × 3 × 3 genotypes. Here the 3 × 3 cross-genotype blocks are nested within each gene block. As a self-correlation matrix, the column IDs are identical to the row IDs. The left panel shows the two sub-networks that were used to drive the simulation, one involving *CDH1* and *CDH10*, the second involving *CDH19*, *PCDH1*, *PCDH10*, and *PCDH17*. *PCHD19* interacted with several genes, but only under certain genotype configurations. This matrix also implies other high order dependencies that are not well shown in this form, but can be observed by tracing from a significant value in a cell, to any other significant value for another gene that occurs in either the same row or column. The number of steps along which such a chain may be followed, defines the number of interacting factors. The correlation matrix re-derived from the output of the simulation (right panel) includes both the spiked-in network and stochastic variation from the simulation, as well as the real biological correlations across genes.

to ordering the parallel coordinates. This approach enabled correct isolation of the spiked-in network, and its modulation of the affected status for individuals, by iterative re-ordering of the coordinates until the affected individuals and the differential properties of their expression levels were clustered within the display. The second tool provided an interactive display of the PLINK-predicted effect of each SNP on each RNA expression level, ordered by genomic locus, and superimposed with the difference in allele frequency between affected and unaffected individuals, as well as a Circos [58]/Mizbee [53] inspired circular display of two locus interactions. Although the entry identified relatively few of the spiked-in SNPs specifically, it did describe many features of the expression interactions that were associated with disease, and many combinations of SNPs that affected expression. It correctly identified the specific effect of the Bonus-question SNP on the gene containing it, but did not arrive at a correct conclusion regarding this gene's overall contribution to affected status. This entry was

overwhelmingly selected by Team Bio as the entry that they would be most comfortable using immediately in their research work, and was selected for the Biology Experts Pick award for the contest. A more in-depth discussion of this entry, from Sakai and Aerts, follows in **Awarded Entries**.

Awarded entries

Three entries were selected by the Judging teams for awards as the Visualization Experts' pick, the Biology Experts' Pick, and the Overall Best Entry. The winning teams were invited to summarize their entries for this manuscript:

Visualization experts' pick: Güter Jäger, Florian Battke, Corinna Vehlow, Julian Heinrich and Kay Nieselt

We present Reveal, a tool for visual analyses of eQTL data. The starting point of an analysis using Reveal is a list of SNPs and genes, and data from a patient cohort covering the presence of the sequence polymorphisms and the expression values of the genes, as well as PLINK results providing information on significant association between SNPs and SNP pairs and differences in expression. A graph is constructed such that each gene in the data set is represented by a node. For each gene the number of significant SNP pairs with one SNP associated with that gene is determined. Nodes of genes with at least one such pair are assigned a unique color, all other nodes are painted using a gray fill.

Edges are added between nodes as follows: Based on the p-values computed for the association between SNP pairs and gene expression, create a triple $< g_i, g_j, g_k >$of genes for each SNP pair with partners in g_i and g_j that is significantly associated with the gene expression of g_k. For each g_k, add an edge between the nodes of g_i and g_j with weight $w = |< g_i, g_j, g_k >|$ and color $c(g_k)$. As SNPs located in, or close to, g_i and g_j can form pairs which influence the expression of different target genes, the graph can contain multi-edges which differ only in color, and possibly in weight. The resulting network is shown in Figure 5.10 (a). All SNPs represented in the network are then displayed in the association viewer iHAT [95] that supports the visualization of multiple sequence alignments, associated metadata, and hierarchical clusterings. Moreover, data-type dependent colormaps and aggregation strategies as well as different filtering options support the user in finding correlations between sequences and metadata. SNPs are colored green if both bases are identical to the reference sequence, yellow if one of the two alleles differs from the reference and red in the case that both alleles differ from the reference. Patient data included affection status (either affected 'red', or unaffected 'white'), is visualized as a meta information column. Furthermore, the gene expression data of the fifteen genes is also

visualized as metadata using a color gradient blue-white-red representing low to high expression.

Next we sorted the column 'affection', resulting in the two groups of 'affected' and 'not affected' patients. Each group was then aggregated, with the aggregate value taken as the specific value observed with the largest relative frequency. The hue of the aggregated SNP value is chosen according to the color scheme for the SNPs described above, and the saturation and value of the color indicates the uncertainty of the aggregate consensus. By visual inspection we then filtered all those SNPs that displayed distinctly different colors between the 'affected' and the 'unaffected' groups (Figure 5.10 (b)).

Biology experts' pick: Ryo Sakai and Jan Aerts
We present an exploratory tool for visual analytics in eQTL data. We performed minimal processing of the provided genotype and phenotype data and instead developed representations for the data in its original form. This decision was based on two factors: First, as the domain expert is already familiar with this type of data, he or she could interpret the visualization without learning new data-related concepts, and therefore could more readily interact and explore new hypotheses; Second, we believe that close interaction and iterative development in collaboration with domain experts is required for developing meaningful processing strategies, and the contest timeline could not accommodate this.

In order to explore and analyze the different aspects of the data, three different visualization modules were created. The first module (Figure 5.11) utilized parallel coordinates [96] defined by the fifteen gene expression levels to visualize each individual as a polyline. Different colors were used to distinguish cases from controls. A histogram was added for each axis/gene representing the distribution of gene expression levels, also stratified by case or control. Simple interactions and filter functions allowed the user to study combinations of different gene expression levels. These interactions included showing only cases or controls, filtering the expression values of any gene by value, and rearranging parallel coordinate axes. In addition, individuals could be filtered by any allele for any given SNP. The second module was targeted at exploring the single locus eQTL analysis data. The display consisted of a matrix of barplots; each bar representing the impact of a single SNP on a specific gene. This module clearly showed that the transinteraction of single SNPs is limited in this dataset, although occasional signals were visible. The third module visualized the two locus data to study the networks of interacting SNPs that affect specific gene expression levels. Association between two SNPs is shown as lines within a circular representation, similar to that used in Circos tool [58]. This representation clearly indicated groups of genes that are part of co-expression networks, including the known co-expression network of *CDH22* and CDH7. Future work includes integrating these three modules into one

cohesive visualization tool and conducting usability studies with domain experts to get insight to iterate both visual and interaction design for the analysis of eQTL data.

Overall best entry: Jesse Paquette and Pek Lum
Our approach focused on visualizing the contest dataset with the Iris software platform (Ayasdi, Inc.), a topology- based exploratory analysis platform for complex datasets (http://www.ayasdi.com). Much as hierarchical clustering produces heatmaps and dendrograms showing how the points (rows) in a data set are related to each other over its dimensions (columns), Iris utilizes topology to capture geometric features in the data and presents relationships between points via interactive network maps. Topological methods often identify structures that elude linear clustering and projection [97, 98, 99]. Our primary goal was to produce a network map in Iris that visualized the effect of the SNPs on the expression of the 15 genes. From the contest-provided data, we produced a matrix M by calculating mutual information (MI) between all pairs of SNPs over all 500 patients. The matrix M was loaded into Ayasadi's Iris Platform [100] and a topological network map was constructed using the program's "Principal SVD lens" with resolution = 30 and gain = 3, and "Correlation Metric" [101].

Figure 5.12 shows the resulting network maps of SNPs produced by Iris. Nodes in each map represent clusters of SNPs and edges indicate clusters that have at least one SNP in common. In other words, every SNP in the dataset can be located in more than one node. The size of each node is proportional to the number of SNPs it contains. Note the starburst shape in the SNP data, with large nodes at the middle and smaller nodes extending towards the tips of the flares. All of the flares in the starburst, except that labeled "Mixed", contain SNPs exclusively from a single locus and are labeled accordingly. For example, all of the SNPs in the CDH10- labeled flare are in the *CDH10* locus. The single-locus flares recover an important pattern in the data: linkage disequilibrium (LD) between SNPs.

The exploratory power of Iris visualization comes from unsupervised construction of the network map, followed by coloring of the map using phenotype values; in this case the phenotypes for the SNPs are relationships with gene expression and disease. Figure 5.12 presents different colorings of the same network map; each color scheme shows how the SNPs relate to disease expression (Figure 5.12 panel A) or individual gene expression (Figure 5.12 panels B-E). The label in the bottom right of each panel indicates the color scheme source. The color of each node represents the mean of the statistic for all of the SNPs contained within. For the color scheme showing relationship to disease (Figure 5.12 panel A), a MI statistic was calculated for each SNP with respect to patient disease status. Larger MI statistics indicate more significant relationships; red nodes contain SNPs with the highest MI vs. disease. For example, in Figure 5.12 panel A, the

flares labeled CHD19 and CHD11 have the highest relationship with disease. For each color scheme showing relationship to gene expression (Figure 5.12 panels B-E), an ANOVA F-statistic was calculated for each SNP with respect to each gene's expression. Larger F-statistics indicate more significant relationships; red nodes contain SNPs with the largest F-statistic vs. individual gene expression. In short, the flares with the warmest coloring are the most interesting. If the disease were simply a function of SNP profiles, then the starburst colored by disease relationships (Figure 5.12 panel A) would implicate SNPs in the *CDH11* and *CDH19* loci (the warm-colored flares) as important influencers of disease. However, given the assumption provided in the contest description that disease is a function of gene expression, and gene expression in turn is a function of SNP profiles, we turned our focus toward the relationships between SNPs and genes.

The network maps in Figure 5.12 panels B-E illustrate the relationships between SNP allelic patterns and gene expression. One can see genes with cis affecting SNPs (in Figure 5.12 panel B the red-colored flare with the highest F-statistic for *CDH19* contains SNPs from the *CDH19* locus), trans affecting SNPs (in Figure 5.12 panel C the red-colored flares with the highest F-statistic for *PCDH17* contains SNPs from the *CDH11* and *CDH5* loci), and very complex expression relationships (e.g. Figure 5.12 panel D). Insights gained from topological network maps with subsequent exploration of color schemes and flare structures can directly lead to hypotheses that can be taken back to the wet lab (or other datasets) and tested. For example, a researcher could identify distinct subsets of SNPs that relate to the expression of *PCDH17* and then design assays to discover which of those were actually affecting *PCDH17* expression, and which ones were simply in LD with them. Alternatively, transposing the SNP $\times$ patient matrix yields a network map of patients. We are extending our methods to other domains such as genome-wide association studies and functional-genomics data to uncover structure and yield new perspectives on these areas.

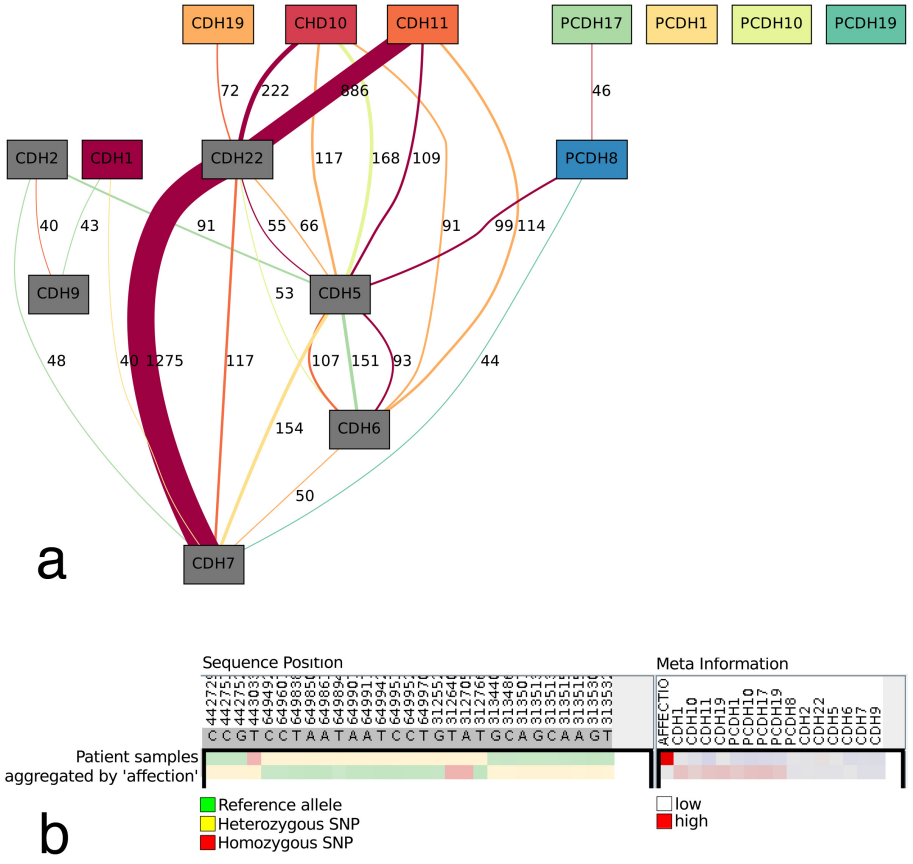

Figure 5.10: The Visualization Experts' Pick. (a) Association gene network ed from all pairs of 3843 SNPs with a significant association ($p < 0.05$, PLINK twolocus results) with the gene expression of the 15 genes and filtered such that only SNP pairs containing at least one highly significant SNP ($R^2 > 0.1$ and $p < 0.05$, PLINK single locus results) remain. All edges with weight $w \geq 40$ are shown. Nodes represent genes, edges represent significant SNP pairs. Genes significantly associated with SNP pairs are colored using a distinct color, genes with no significant association are drawn with gray fill. Each edge conveys four pieces of information: An edge e of weight w starting in node s, ending in node t and drawn with color c represents w SNP pairs, where each of them has one SNP in gene s and one in gene t. These SNP pairs are significantly associated with the expression of the gene whose node is filled with color c; (b) Aggregated iHAT visualization of 29 visually selected SNPs where the 'affected' and 'not affected' groups display different colors.

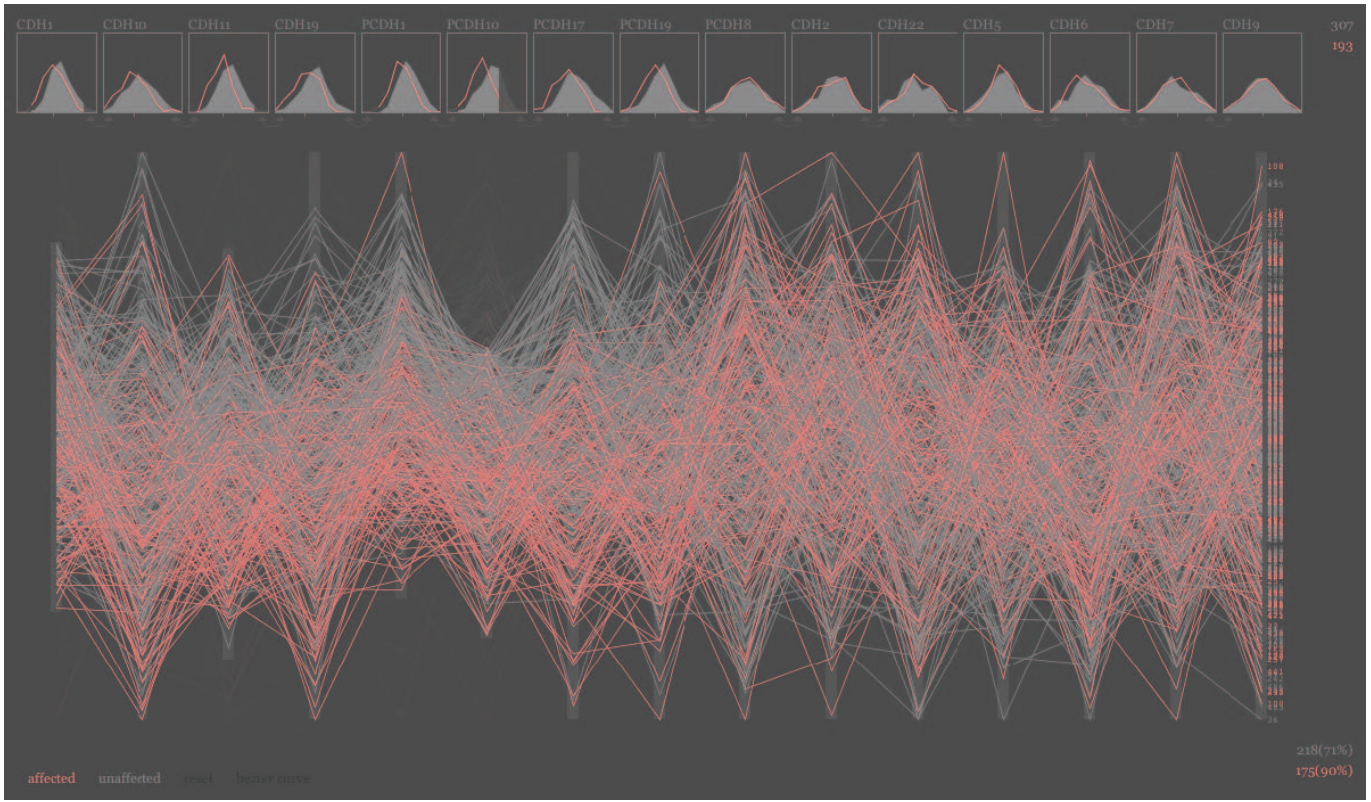

Figure 5.11: The Biology Experts' Pick. Parallel coordinate display of gene expressions per individual. Vertical axes represent expression level for a given gene; horizontal polylines across the display represent each individual. Individuals are stratified in case (pink) versus control (grey). At the top of each vertical axis a histogram displays the distribution of expression levels of that gene over all individuals, stratified by group. The data for genes 1, 3, 5 and 6 are filtered for high and/or low values in this figure.

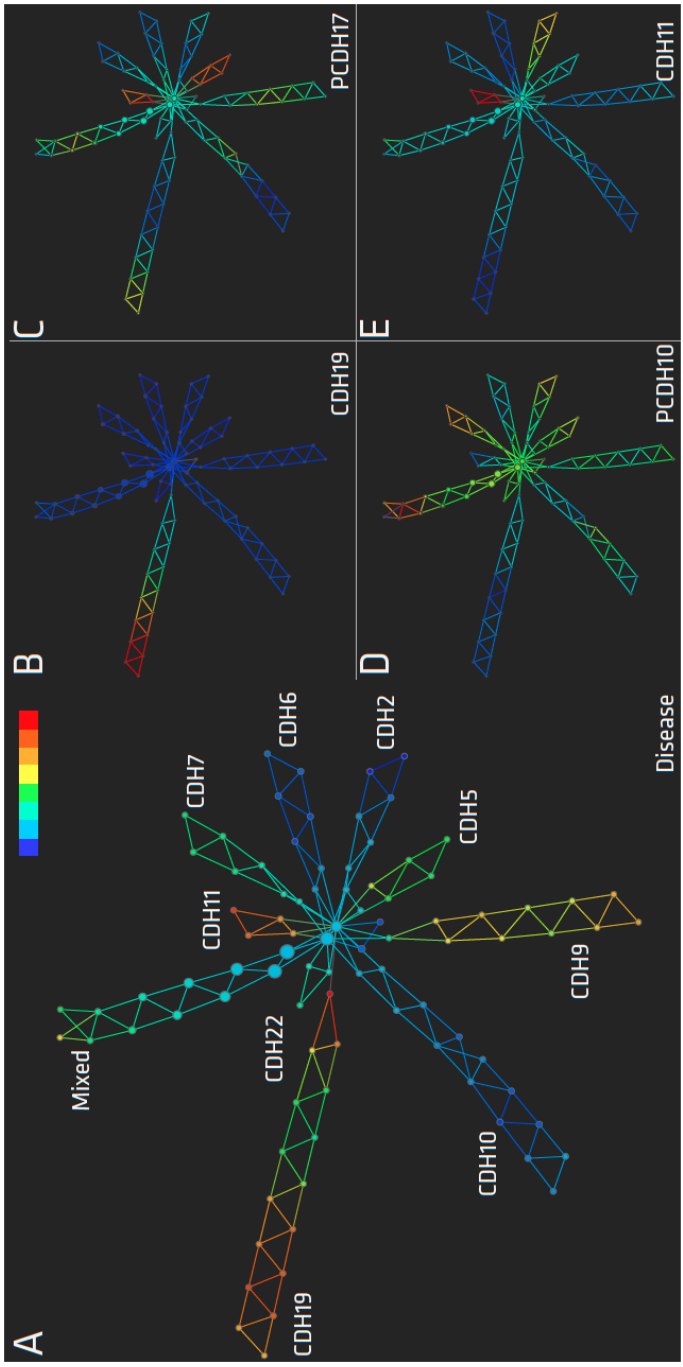

Figure 5.12: The Overall Best Entry. A topological network map of SNPs produced by Iris. Each node represents a cluster of SNPs and nodes are connected with an edge if they have any SNPs in common. The starburst shape indicates subgroups of SNPs with distinct linkage disequilibrium patterns in the data set. A) Each flare of the starburst contains SNPs from a single locus and is labeled accordingly, except for the "Mixed" flare. The nodes are colored by SNP mutual information with disease. Higher mutual information values are colored red and indicate a stronger relationship. B) The nodes are colored by SNP ANOVA F-statistic with expression of *CDH19*. Higher F-statistics are colored red and indicate a stronger relationship. The flare with the red tip contains SNPs from the *CDH19* locus; see label in A. C) The nodes are colored by F -statistic to expression of *PCDH17*. D) The nodes are colored by F-statistic to *PCDH10*. E) The nodes are colored by F -statistic to *CDH11*.

Chapter 6

Interaction Design

6.1 Interaction

Interaction is concerned with how the user changes the way the data is presented. As shown in Figure 6.1, an interaction is initiated by the user based on their tasks and any hunch or insight from the current state of visualization. For example, the user may select an item to highlight, reveal more information, move the location of the item, or change a filter setting. The design space of possible vis idioms is extended further with possible combinations of visual encoding design and interaction design [7]. In this chapter, a design principle ("**Get it right with clicks**") for prototyping of custom visualization tools is

introduced via three design studies. These studies highlight how interaction influenced both the analysis and the design process.

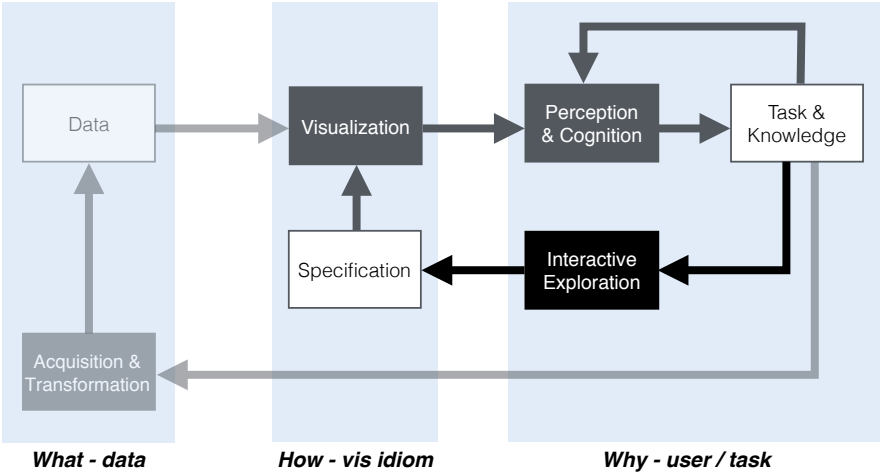

Figure 6.1: A framework for vis design, highlighting the role of the interaction.

The concept of **Get it right with clicks** advocates the use of mouse clicks as a primary interaction method to prototype for four reasons. First, the event handling of a click action is easier to program. Second, most users are already familiar with a click action. Third, unlike mouse hovering events, a click is intentional rather than accidental. Fourth, mouse click events are usually sufficient to keep the flow of exploratory analysis. For example, a typical data analysis in R is disruptive because the analyst has to go back to the script to make any changes in the visualization generated. This disruption without a graphical user interface is tolerable if you are just refining a figure design, but the disruption of the flow is detrimental when the main goal is exploratory. In an exploratory analysis, often there are many combinations of attributes or parameters to test. By allowing the user to change parameters interactively with reasonable latency on display [102], it can speed up and improve our ability to explore the data more thoroughly and efficiently.

This chapter includes three design studies: **Brain Constellation**, **TrioVis**, and **Dendsort**. *Brain Constellation* was developed for the BioVis 2014 data contest that focused on the domain of resting state functional connectivity networks. *TrioVis* is an interactive visualization tool developed to gain better intuition about the coverage and the variant frequency setting for analysis of exome sequencing of parent-child trios. Although two different domains, both tools leverage the simple click interaction to explore filter threshold

parameters to understand the underlying data structure and perform analysis tasks. *Dendsort* is an R package implementing leaf ordering heuristics. The research of dendrogram and cluster heatmap visualization led to the development of this leaf ordering methods, and interaction designs in earlier prototypes were instrumental in designing these leaf ordering algorithms.

6.2 Case Study: Brain Constellation

Collaboration:

Nico Verbeek[1] and Jaak Simm[1]

N.V. and J.S. analyzed the data and disscussed the design of the Brain Constellation with the candidate.

[1]*KU Leuven, ESAT - STADIUS, Leuven, 3001, Belgium*

The BioVis 2014 data contest focused on the domain of resting state functional connectivity networks (rs-fMRI network) [103]. These networks were derived from functional magnetic resonance imaging scans of human subjects, which measured the blood oxygenation level dependent (BOLD) activity over a period in different regions of the brain. The time-series data were pre-processed to weighted adjacency matrices with each row and column corresponding to a region of interest (ROI). The values corresponded to the strength of coupling between two anatomical regions (Figure 6.2). The contest provided two analysis tasks: 1) to characterise most consistent and variable properties of the network across the population of subjects provided, 2) to classify unknown networks to the corresponding subject network. To address these tasks visually, an interactive visual analytics tool, called Brain Constellation, was developed. This vis tool incorporated a simple user interaction following the **Get it right with clicks** concept, and there were two key design choices in the data analysis that made this tool effective for the tasks proposed.

The first key design choice was the abstraction of three-dimensional physical data into a two-dimensional representation of the 169 brain ROIs. In the neuroscience literature, there are two common types of visualization used to encode the BOLD activity: the volumetric rendering of 3D brain structure and the coronal, sagittal or transverse slice of the brain [104, 105]. With these vis idioms, studying the whole brain structure requires some form of user interaction to explore the regions that are hidden from the view. These visualizations may be sufficient for a communication purpose, and only if the message to communicate has already been identified. However, the exploratory analysis would require the user's ability to rotate the 3D structure or to adjust the position of the planes to study the whole brain structure. Also, these views are not suitable for comparing multiple samples due to constraints of display size and its associated high cognitive load.

Therefore, the main motivation was to develop an alternative representation of ROIs that supports comparison between samples. First, three-dimensional

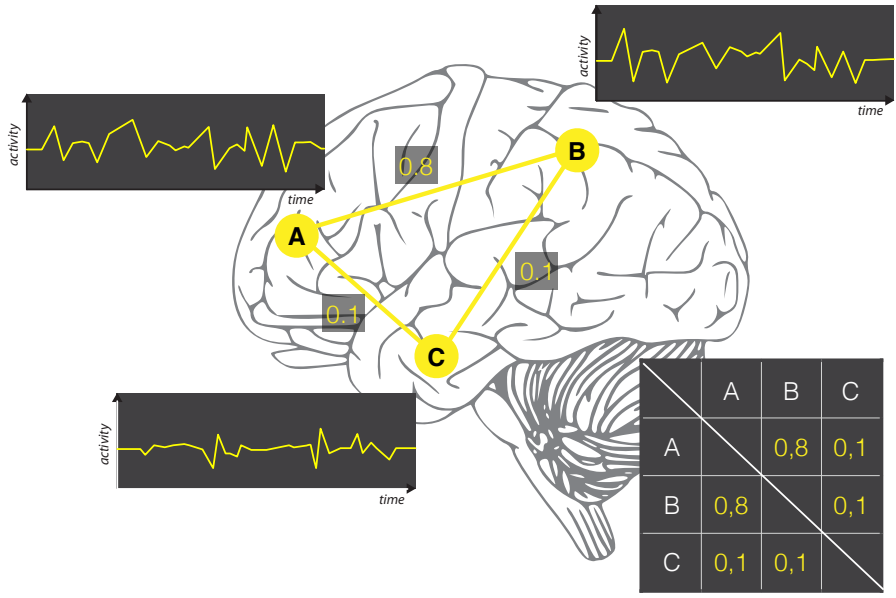

Figure 6.2: A illustration of how the correlation between regions of interest (ROI) is measured based on the blood oxygenation level-dependent (BOLD) activity over a period of time. The point A, B and C are examples of ROI.

coordinates of ROI were projected onto a two-dimensional orthogonal plane by performing Principal Component Analysis (PCA) (Figure 6.3). This plane consisted of the first two principal components. Although this dimensionality reduction lost 24 percent of variance explained by the third principal component, the resulting two-dimensional plane preserved as much anatomical structural information as possible and created a template with the least number of overlapping ROIs. This two-dimensional template allowed comparison of multiple networks while maintaining most of the anatomical relevance.

The second key design choice was the calculation of "correlation of correlation patterns" per ROI to compare the similarity between two networks. This ROI-wise correlation value was used to classify an unknown network to a known network. A schematic diagram of this approach is shown in Figure 6.4. By setting a threshold for the correlation value, only those ROIs with similar connectivity patterns were highlighted in each representation of a network. Then, the best candidate for classification was selected based on the total count

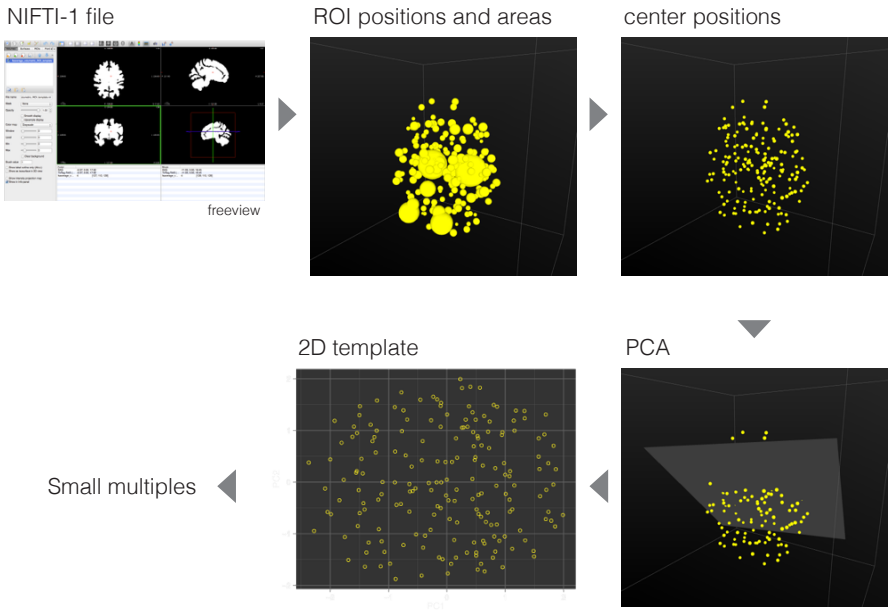

Figure 6.3: A sequential steps to create a two-dimensional template for small multiples. From the anatomical template with ROI annotations, the positions and the mass for each ROI were determined, and then the three-dimensional coordinates for each ROI were derived from their centre of mass. Principal Component Analysis was used to find the plane to project these three-dimensional coordinates. The first two principal components captured 75% of the variance in data and were used as a template for small multiple visualizations.

and the position of highlighted ROIs (Figure 6.5).

Because the optimal filter threshold setting varied from sample to sample, a simple clicking interaction was implemented to explore and find the most suitable threshold setting. Although one may argue that this is an optimisation problem and can be automated, the cost of implementing a simple user interaction is much lower than that of implementing an appropriate metric and a grid search. Arguably, these interactive features to adjust the thresholds would be useful for developing and evaluating such a computational approach. An automated solution would save the user's time in the long run. For our purpose, the simple click mouse interaction was sufficient, and most importantly, the result showed that it is indeed effective as we were able to classify all the networks perfectly

for the second analysis task.

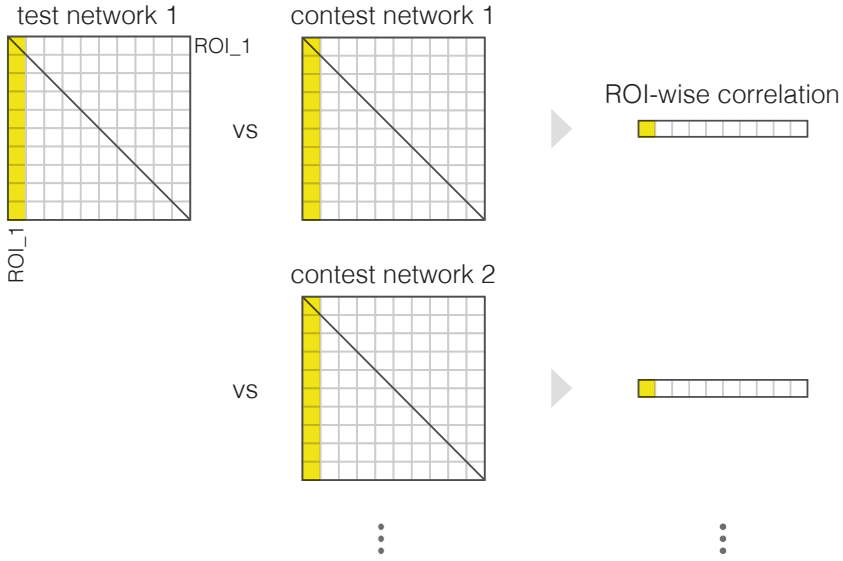

Figure 6.4: A schematic diagram of ROI-wise correlation for the classification task. How a selected brain region correlates with the rest of brain regions is compared between the known and unknown samples. This calculation yields a vector of ROI-wise correlation values.

With *Brain Constellation*, we were able to address both analysis tasks defined by the contest and achieve the perfect classification result via visual analysis. As a result, our entry won the *Overall Favorite Data Contest Award* at BioVis2014 (ISMB 2014). The comments from reviewers were positive and scored highly because of its simplicity and effectiveness. A common suggestion for future work was to link the nodes on the two-dimensional template to the three-dimensional structure of the brain.

The design process also illustrates that a vis design often, if not always, involves evaluation of the pros and cons. One vis idiom may be better at showing a particular property or a pattern at the cost of possibly obscuring other attributes. As reviewers commented, the emphasis of Brain Constellation is on the comparison of networks, rather than the anatomical accuracy as seen in the conventional volume rendering or scan images. A careful analysis of the tradeoffs in visual encoding and interaction design is the essence of data visualization research.

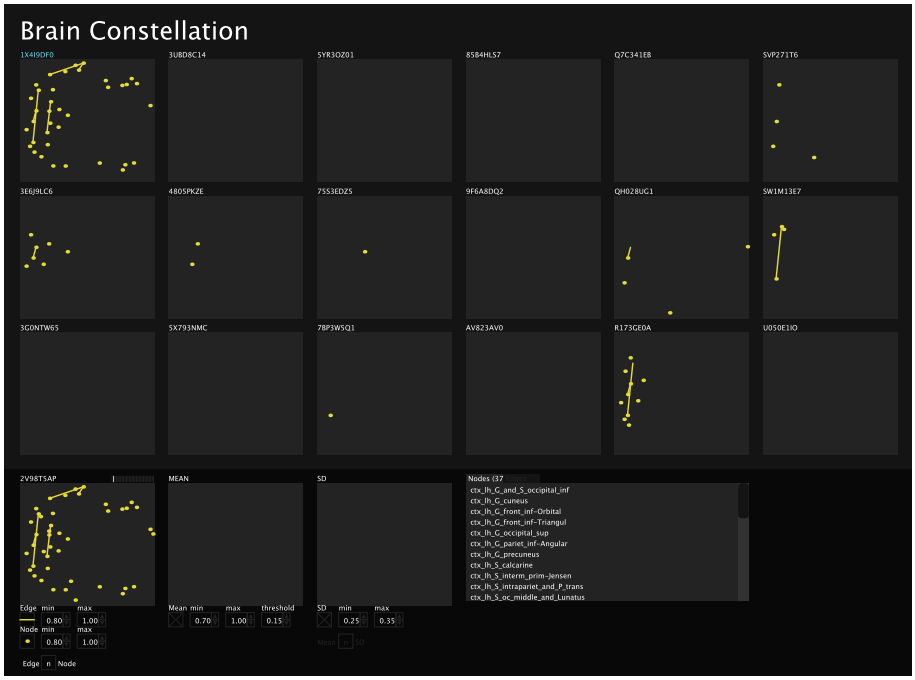

Figure 6.5: The user interface of Brain Constellation for the classification task. The bottom left panel shows the unknown network, which is compared against 18 known subject networks. This view shows the matching network on the top left.

This design study demonstrates a simple mouse click interaction is sufficient to empower analysts in the visual analytic process of characterising and classifying brain networks. The interface employed small multiples of two-dimensional, orthogonal projection of brain ROIs. Although the tool addressed the contest tasks well, these analysis tasks were "toy" tasks devised to enable contest organisers to score entries. Hence, these tasks were somewhat disconnected from the real-world research questions.

6.3 Case Study: TrioVis

Collaboration:

Mala Isrie[1]

M.I. was the domain expert and the target user of the TrioVis.

[1]*KU Leuven, Department of Human Genetics, Leuven, Belgium*

Publication: Section 6.5

TrioVis is a visualization tool developed to gain a better understanding of the coverage and variant frequency settings for filtering genomic variants from exome sequencing of parent-child trios. The main motivation, or the target task (Why), originates from an interview with a geneticist studying Mendelian monogenic disorders. During the interview, we discussed and went through steps in their typical analysis process as well as reasonings at each step for variant discovery. The task we agreed to focus on for a design study was about finding the optimal filter thresholds for read coverage for trio cases.

A trio case consists of sequencing results from both parents and the child. Typically, a researcher tries different threshold settings until they obtain a list of variants that is manageable in size. The researcher may choose to set a stringent or lenient setting depending on the sequencing quality and the read coverage. Because each sample in a trio case may have different sequencing quality, choosing a single threshold value to apply to all samples is not necessarily ideal for keeping as many true positives and removing as many false positive variant calls as possible. The process of setting thresholds was essentially a trial-and-error basis, trying different settings based on educated guesses. Hence, we decided to develop a visualization system to support the decision-making process of setting filtering thresholds.

A key aspect of interface design of TrioVis is the tabular structure to categorise variants based on their inheritance pattern (Figure 6.6). With small multiples of overlaying histograms based on the coverage, TrioVis compares distributions of variants for each sample per inheritance pattern. Based on the Mendelian inheritance laws, we categorise each variant plausible or improbable. For example, if both of parents are a homozygous reference for a locus, it is unlikely that the locus is a homozygous alternative for the child, thus this variant is most likely to be false positive. The simple bar graph summarises the total count of plausible and improbable variants to evaluate the ratio of true positive and false positive estimates.

After a few iterations, we also implemented a computational approach to finding an "optimal" threshold based on the best-weighted average of the precision and recall (f-score). However, even with this "optimal" setting, there was room for human judgement to decide whether to make the setting stringent or lenient based on the user's domain knowledge or experience. The real value of TrioVis is that it makes the filtering process transparent and dynamic, where a user can assess settings by just clicking to adjust settings and observing its live feedback. This feedback loop of user interaction is illustrated in Figure 6.1. The features of TrioVis are described in more detail in Section 6.5.

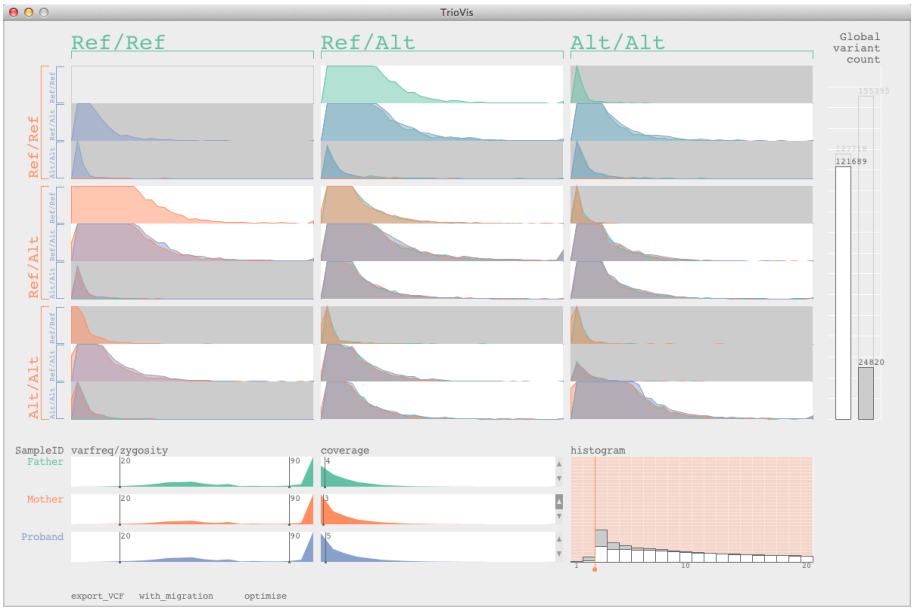

Figure 6.6: The interface of TrioVis.

6.4 Case Study: Dendsort

Collaboration:

Sheila Reynolds[1], Vésteinn Þórsson[1], Dick Kreisberg[1], and Ilya Shmulevich[1]

S.R. and V.T. were the domain experts and the target users. All collaborators contributed to the design of interactive prototypes.

[1]*Institute for Systems Biology, Seattle, WA, USA*

Publication: Section 6.6

Dendsort is an R package developed to improve interpretability of dendrogram and cluster heatmap visualizations by reordering the hierarchical structure. In the development process, there were two preceding interactive visualization prototypes that were instrumental in coming up with the idea and implementing an automated solution. The following describes how the design study started and how each interactive visualization prototype supported the exploration of design space.

The project started during the visit to the Shmulevich lab at the Institute for Systems Biology in Seattle in the fall of 2013. During this visit, I had an opportunity to work with research scientists who were part of The Cancer Genome Atlas (TCGA) project. The research group had developed a statistical method to integrate disparate data types, such as clinical diagnosis, treatment history, histological diagnosis, gene expression, and sequencing data [106]. This integrative analysis resulted in a large table of correlation values between pathways and clinical cohort groups. Typically, this table was visualised as a cluster heatmap to study and characterise subtypes of a cancer type.

One key observation was made while observing a researcher analysing a large cluster heatmap. The analyst was holding a blank sheet of paper against her computer screen while tilting her head sideways to read the vertical labels. There were two key insights from this observation. First, the heatmap was so large and dense that the analyst needed a "ruler" to read the corresponding labels of clusters identified. Second, the labels were critical for understanding and characterising the clusters, but the labels in their current form were hard to read. Many researchers are very creative in finding a solution to support their analysis, and careful observation of their analysis processes can provide useful insights into their analysis tasks or limitations of current visual representations.

Intrigued by the observation, a few sessions of discussion with the analysts followed and an interactive heatmap visualization prototype (Figure 6.7) was

developed. In this interactive heatmap, the dendrograms on top and the left were interactive, where a click on a branch introduce a gap in the heatmap visualization. This insertion of horizontal or vertical gaps was inspired by [107] and was analogous to the use of a blank sheet of paper from the observation. The gaps helped to isolate clusters in heat maps without occluding the view as with using the paper.

The dendrogram representation also incorporated the result of multi-scale bootstrap resampling [108] to emphasise those "stable" clusters. Those stable clusters are shown in thicker lines in dendrograms. The prototype also allowed the user to select a cluster and review the information about the selection in the lists on the bottom. Because each column in the dataset represented a pathway (a set of genes), an order list of genes based on the number of occurrence in the selected pathways was shown in the second list. To examine the selected set of genes further, the prototype linked to the underlying gene expression dataset and generated a cluster heatmap based on the gene expression data. With a single click, the prototype retrieved the data, performed hierarchical clustering, and presented a cluster heatmap in a pop-up window (Figure 6.8).

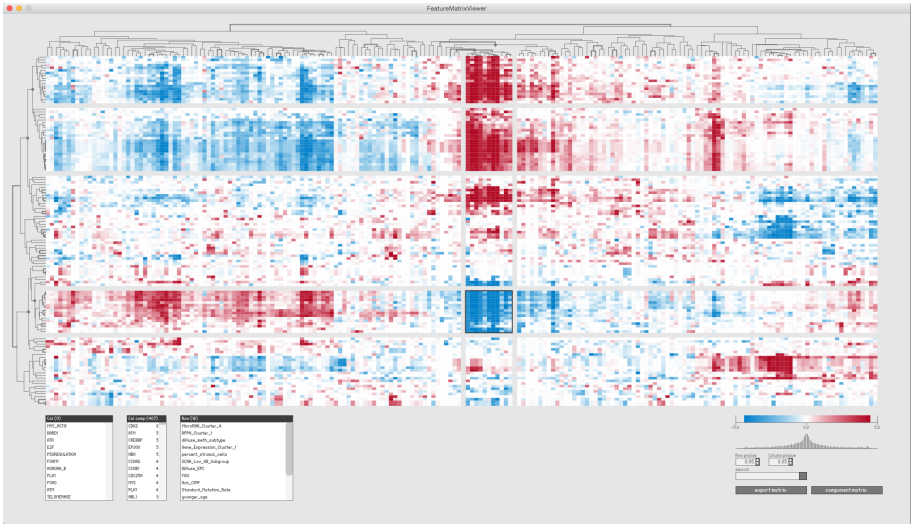

Figure 6.7: An interface of the first interactive heatmap prototype.

This prototype had additional features to support the visual analysis. Although a heatmap is a compact representation of tabular data, our ability to decode the quantitative value from a colour scale is limited [109, 15]. Thus, a colour slider to adjust colour mapping was implemented to highlight only those values at either end of the distribution (Figure 6.9). By simplifying the heatmap

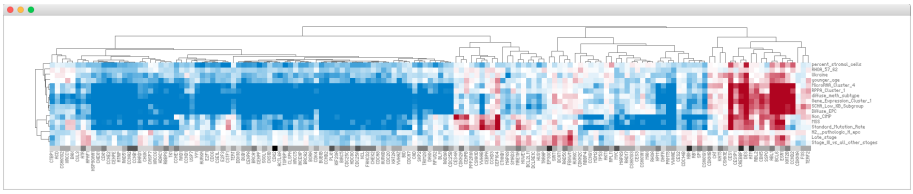

Figure 6.8: A cluster heatmap generated based on the selection of a cluster. The last row in heatmap shows the number of occurrence in the selected pathways in a gray scale.

visualization, the patterns of cohorts were more readily perceived. In addition to the cluster analysis mode, the prototype had a detail view mode, in which the user could hover over an individual cell to get their detail information (Figure 6.10). By placing the label right next to the mouse cursor, the distance between a selected cell and its associated information was much shorter, hence allowing a more efficient examination of the data.

Figure 6.9: The color mapping scale is adjusted to highlight those values at each end of the value distribution.

The presentation of this interactive prototype led to a discussion about a specific property of a dendrogram: the leaf ordering. A dendrogram is a representation of a binary tree structure, in which each branch can be rotated around its axis without changing the meaning of the hierarchy. To explore this property further, a prototype with interactive dendrograms was implemented (Figure 6.11). When

Figure 6.10: The detail view mode of the prototype.

a user clicked on a branch, it swapped the positions of sub-clusters and the heatmap visualization was updated immediately. After exploring different permutations of leaf ordering, we learned that we can design an algorithm to reorder a dendrogram to improve the interpretability of heat map and dendrograms. The result of reordering is shown in Figure 6.12. Encouraged by the results, we implemented an R package, called **dendsort**, and then evaluated its performance as described in Section 6.6.

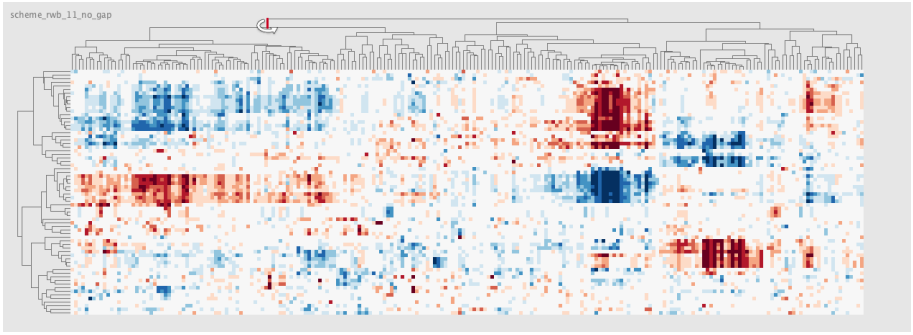

Figure 6.11: The interface of a prototype with interactive dendrograms to rotate along the selected axis.

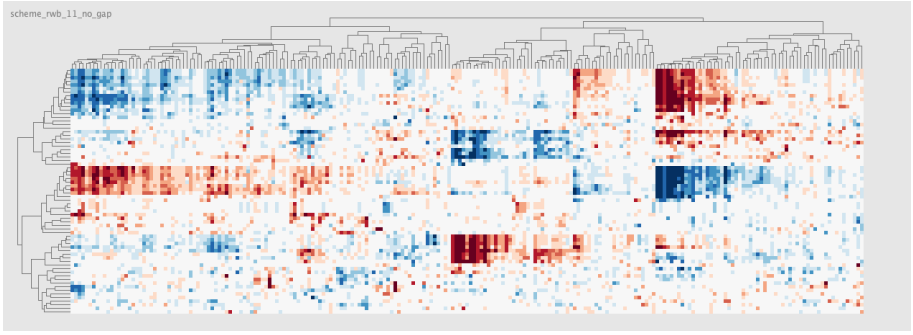

Figure 6.12: The result of applying heuristics to reorder the dendrograms.

Besides the leaf ordering methods, we also considered another design solution to improve the interpretability of cluster heat maps by introducing gaps that encode the distance between two adjacent leaves. This concept extended the work by [107] and addressed a common misinterpretation of distance between two adjacent nodes in dendrograms. Figure 6.13A shows selected cities in Europe, and the geographic (Euclidian) distance between each city was used as an input for agglomerative hierarchical clustering with the complete linkage algorithm. Figure 6.13B shows a resulting dendrogram. A common mistake in interpretation of a dendrogram is to assume two items are similar just because they end up next to each other in the resulting linear order. For example, Stockholm and Lisbon are next to each other in the linear order. However, we know that they are not close to each other geographically. In a dendrogram representation, the actual distance between two cities is only encoded in the height of the branch at which two subclusters meet. Figure 6.13C shows the same dendrogram but with gaps to encode the distance between two adjacent

nodes in the dendrogram. The clusters of cities are clearer, which makes the dendrogram structure easier and more intuitive to interpret. The methods were implemented and distributed as an R packaged named **gapmap**.

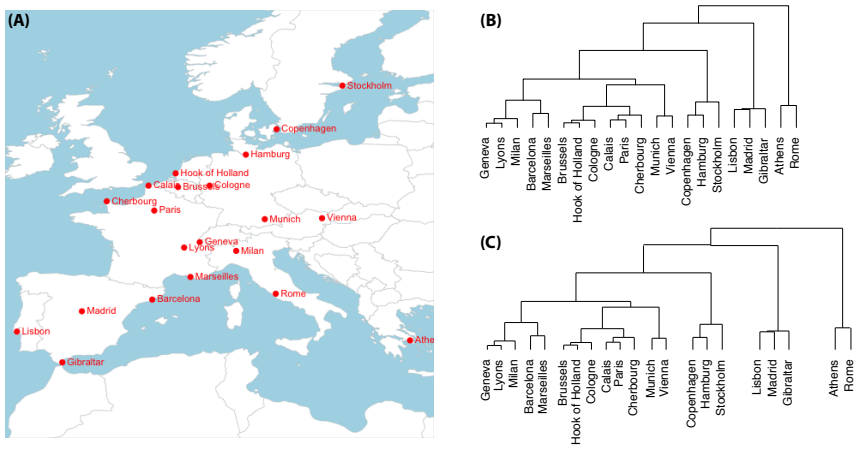

Figure 6.13: Comparison of dendrogram structures with and without gaps using the distance between European cities. (A) A map of Europe with selected cities. (B) A dendrogram structure without gaps. (C) A dendrogram structure with gaps.

This design study demonstrates how a visualization project can inspire a development of a computational approach or an algorithm. The leaf ordering method is a computational solution, but it would not have been realised without the interactive vis prototypes and discussions with collaborators. This project also illustrates how a prototype designed for a specific task can create design solutions to address different tasks. Hence, a vis prototype is not just for the analysis tasks, but also for ideation of new visualization techniques.

6.5 TrioVis: a Visualization Approach for Filtering Genomic Variants of Parent-child Trios

R. Sakai, A. Sifrim, A. Vande Moere, and J. Aerts, "TrioVis: a visualization approach for filtering genomic variants of parent-child trios.," *Bioinformatics*, pp. 1–2, Jun. 2013. (Reprinted with permission. License Number: 3691241271769.)

6.5.1 Summary:

TrioVis is a visual analytics tool developed for filtering on coverage and variant frequency for genomic variants from exome sequencing of parent-child trios. In TrioVis, the variant data are organised by grouping each variant based on the laws of Mendelian inheritance. Taking three Variant Call Format (VCF) files as input, TrioVis allows the user to test different coverage thresholds (*i.e.* different levels of stringency), to find the optimal threshold values tailored to their hypotheses, and to gain insights into the global effects of filtering through interaction.

6.5.2 Availability:

Executables, source code and sample data are available at https://bitbucket.org/vda-lab/triovis/. Screencast is available at http://vimeo.com/user6757771/triovis.

6.5.3 Introduction

Recent advances in massively parallel sequencing technologies, especially sequencing of the entire protein-coding portion of the genome (exome), have introduced new strategies for identifying Mendelian disease genes [110]. Analysis of parent-child trios is one of the strategies for identifying single pathogenic mutations amongst the thousands to millions of genomic variants. By sequencing the patient as well as his or her parents, variants can be filtered based on consistency or inconsistency according to the laws of Mendelian inheritance.

Although filtering based on inheritance pattern appears straight-forward, distinguishing true variation from artefacts and false negatives while retaining sensitivity is a challenging task because of the sequencing error rate and the interdependency of sequencing quality for multiple samples. A previous study

[111] reported that more than 70 percent of Mendelian inconsistencies were found to be false negatives due to the failure to call the germline variant in either parent sample in search for *de novo* mutations. Similarly, we found that the majority (77 %) of variants was consistent with the Mendelian laws when we analysed those variants that are in common between the exome sequencing and a SNP genotyping array for a trio-case (data not shown). One of the metrics commonly used to filter variants is the depth of coverage. Researchers we interviewed adjust the coverage threshold based on the overall coverage and their intuition without any visual aids. The optimal coverage thresholds also depend on other factors, such as the suspected type of mutation, whether somatic or inherited, and the stringency of analysis. Although finding the optimal coverage threshold can be automated to some extent, it still requires fine adjustments of the filtering setting for variant discovery.

We present a visual analytics tool, TrioVis, designed to help the analytical reasoning process of setting coverage thresholds to filter variants from parent-child trio sequencing experiments. It visualises variants in a structured table and provides interactive visual interfaces to let the researcher dynamically and interactively test different threshold settings and change levels of stringency.

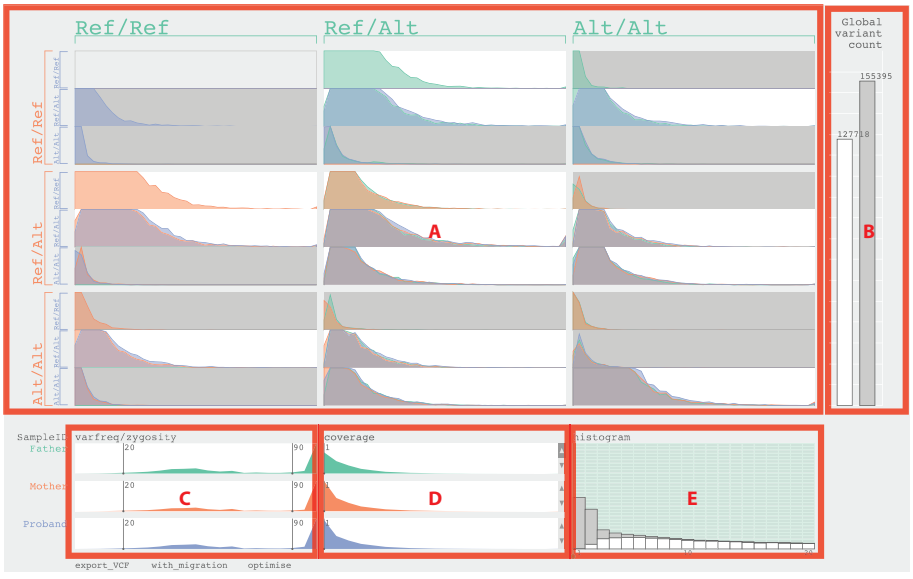

Figure 6.14: The user interface of TrioVis and five sections labeled in red for parent-child trio data from exome sequencing by Illumina HiSEQ 2000 are used. (A) The main table. (B) The global variant count bar graphs. (C) The variant frequency sliders. (D) The coverage sliders. (E) The histogram view showing the father sample.

6.5.4 Features

TrioVis is a standalone, desktop application developed in Processing [70], an open source programming language and integrated development environment (IDE), based on Java, and is available for Linux, Mac OS X and Windows. It loads three separate Variant Call Format (VCF) files, and sample VCF files were generated using the GATK Unified Genotyper [112]. It requires the AD (Depth Per Allele By Sample) field, which includes the unfiltered count of reference (REF) or alternative (ALT) reads. Based on these read counts, variant frequencies for each variant are calculated. A sample data set, generated from the BAM files of Utah residents with Northern and Western European ancestry (CEU) trios from the 1000 Genome Project [113], is available for download.

The user interface consists of five sections: the main table Figure 6.14A), the global variant count bar graphs Figure 6.14B), the variant frequency sliders Figure 6.14C), the coverage sliders Figure 6.14D) and the histogram view Figure 6.14E). Each section focuses on a specific aspect of trio data, and offers specific interactive features to calibrate the thresholds. Father, mother and child are colour-coded in green, orange, and blue, respectively.

The *main table* Figure 6.14A) is divided into small multiples based on the pattern of inheritance. Each block consists of three histograms, conveying the distribution of variants based on the read depth per sample. The background colour of each block is determined by whether it is consistent (white) or inconsistent (grey) with the laws of Mendelian inheritance. The *global variant count bar graphs* Figure 6.14B) represents the total counts of variants based on whether or not it is consistent. By changing coverage settings, the researcher aims to minimise the number of inconsistent calls while keeping the number of consistent calls high.

The *variant frequency sliders* Figure 6.14C) visualise the distribution of variants based on variant frequency values. These sliders can be used by the researchers to adjust the ranges for variant frequency for genotyping variants for that sample. By default, any variants with variant frequency higher than 90 are considered alternative homozygous and any variants with variant frequency between 20 and 89 are considered alternative heterozygous. Any variants below 20 are filtered out. The *coverage sliders* Figure 6.14D) set the coverage thresholds for each sample individually. These sliders also represent the distribution of variants based on coverage values. Lastly, the *histogram view* Figure 6.14E) represents the distribution of consistent and inconsistent variants in stacked bar graphs with coverage values between 1 and 20 for the selected sample. Hovering the mouse over the stacked bar graph highlights cells in the main table, showing where these variants are represented. This view aids the researcher to calibrate

the coverage threshold for the selected sample.

The variant data can be investigated under two assumptions: with the "migration" assumption, any variant below the coverage threshold is considered homozygous reference; when this assumption is inactive, variants below the coverage threshold are considered invalid and discarded from the combined set of variants. Filtered results can be exported using the "export VCF" button and saved as VCF files. The researcher can also select specific blocks to export variants of a specific inheritance pattern (*i.e. de novo* mutations and recessively inherited variants) for further analysis. The "optimise" function finds the best-weighted average of the precision and recall (f-score) based on the number of filtered consistent and inconsistent variants, providing the user a good initial setting for further investigation and adjustment.

6.5.5 Conclusion

TrioVis provides an interactive interface and optimisation function to calibrate coverage thresholds based on Mendelian inheritance laws for parent-child trio cases. By visualising variants in a novel table layout based on the inheritance laws, it allows the researcher to gain insights into the global effect of filtering in the context of trio analysis. The resercher can export the filtered result as VCF files for subsequent analysis to annotate variants to genes, using annotation tools such as Annotate-It [114] and Annovar [115]. Future work includes improving the optimisation algorithm, and integration of this tool into functional annotation tools such as Annotate-It and Galaxy [116].

Acknowledgement

The authors wish to thank Mala Isrie for providing trio sequencing data for developing and testing this visualization tool.

Funding: This work was supported by iMinds [SBO 2012], University of Leuven Research Council [SymBioSys PFV/10/016, GOA/10/009] and European Union Framework Programme 7 [HEALTH-F2-2008-223040 "CHeartED"]

6.6 dendsort: Modular Leaf Ordering Methods for Dendrogram Representations in R

R. Sakai, R. Winand, T. Verbeiren, A. Vande Moere, and J. Aerts, "dendsort: modular leaf ordering methods for dendrogram representations in R," *F1000Research*, vol. 177, 2014.

6.6.1 Abstract

Dendrograms are graphical representations of binary tree structures resulting from agglomerative hierarchical clustering. In Life Science, a cluster heat map is a widely accepted visualization technique that utilizes the leaf order of a dendrogram to reorder the rows and columns of the data table. The derived linear order is more meaningful than a random order, because it groups similar items together. However, two consecutive items can be quite dissimilar despite proximity in the order. In addition, there are 2^{n-1} possible orderings given n input elements as the orientation of clusters at each merge can be flipped without affecting the hierarchical structure. We present two modular leaf ordering methods to encode both the monotonic order in which clusters are merged and the nested cluster relationships more faithfully in the resulting dendrogram structure. We compare dendrogram and cluster heat map visualizations created using our heuristics to the default heuristic in R and seriation-based leaf ordering methods. We find that our methods lead to a dendrogram structure with global patterns that are easier to interpret, more legible given a limited display space, and more insightful for some cases. The implementation of methods is available as an R package, named "dendsort", from the CRAN package repository. Further examples, documentations, and the source code are available at [https://bitbucket.org/biovizleuven/dendsort/].

6.6.2 Introduction

Agglomerative hierarchical clustering (HC) is one of the classic and yet still very popular cluster analysis methods in data exploration [117, 89]. Its implementation is widely available and execution of the clustering requires only a few settings, such as a choice of distance metric and linkage algorithm [118]. The clustering process begins with individual input elements as singleton clusters and successively merges a pair of most similar clusters until only one

cluster remains. The dissimilarity, or the distance, between two clusters is defined by a distance metric and updated by a linkage algorithm. The output of HC is typically represented in a form of a binary tree, called a dendrogram. In a dendrogram, the similarity of two clusters is encoded in the height of the branch where two clusters merge. Two very similar elements are merged in the early stages of clustering, thus the height of the branches between these elements is relatively small. The dissimilarity between two clusters increases with each successive merge, resulting in a binary hierarchical structure with a monotonic property [119]. Therefore, a dendrogram represents both cluster-subcluster relationships as well as the order in which the clusters were merged [120].

There are two unique uses of a dendrogram in exploratory data analysis. First, clusters of input elements can be inferred from the subtree structures below a certain threshold by "cutting the tree." It is an advantage of hierarchical clustering that this threshold value can be adjusted based on domain-specific knowledge to result in clusters of different sizes. Second, a linear order of observations (rows) or attributes (columns) of an associated matrix can be derived. This linear order of observations is typically used to reorder the columns or rows of the data matrix. Then, the matrix is visualized as cluster heat maps [117], where dendrograms and heat map visualizations are coupled (Figure 6.15).

The linear order derived from a dendrogram is more meaningful than a random order, as it groups similar items together [121, 107]. However, two consecutive items in this order are not necessarily similar, since these leaves could belong to different subtree structures, or simply be quite distant from each other. This is a common misinterpretation of a dendrogram: one may expect similarity between two input elements based on the proximity in the leaf order [122, 123]. In addition, there are 2^{n-1} possible orderings given n input elements, because the orientation of clusters at each merge can be flipped without affecting the underlying hierarchical structure, thus rendering a unique optimization challenge.

To address the misinterpretation of dendrograms and the optimization problem, a number of methods have been proposed to rearrange the structure of a dendrogram. Gruvaeus and Wainer [124] proposed a method (GW) to order leaves such that two singleton clusters at the edge of adjacent subtrees are most similar, given the constraint of the binary tree structure. Bar-Joseph et al. [121] proposed a method, called the optimal leaf ordering (OLO), to maximize the sum of the similarity of any adjacent elements in the ordering. Similarly, Chae and Chen [125] proposed a method for ordering by minimizing the bilateral symmetric distance between two adjacent clusters. All these methods aim to homogenize the linear order in one way or another and are evaluated in terms of either a loss function, such as the Hamiltonian path length, or a merit function,

such as the number of anti-Robinson events [126].

Even though these seriation-based leaf ordering methods exploit the binary tree structure to reduce the number of permissible permutations, these methods have shortcomings. First, they homogenize and optimize the distance between items in the linear order, and this still encourages the common misinterpretation of dendrograms, reading a dendrogram horizontally. Second, the dendrogram structure is only a means to reduce the number of permissible permutations, and the graphical representation of the resulting dendrogram obscures the intrinsic properties of the hierarchical clustering result, such as the cluster-subcluster relationship and the order in which clusters are merged.

In the biological domain, Eisen et al.[97] have introduced and established a cluster analysis method for high throughput gene expression data using cluster heat maps. The method includes leaf orderings by weighting genes based on genome coordinates or the average expression level. The resulting linear order is more meaningful in terms of biology, but the method requires prior knowledge or additional information for the weighting.

In this paper, we present leaf ordering heuristics, named modular leaf ordering (MOLO), to address the aforementioned shortcomings by constructing a dendrogram that reflects a) the monotonic order in which clusters are merged and b) the nested cluster relationships. We compare dendrogram and cluster heat map visualizations created using our heuristics to the default heuristic in R and seriation-based leaf ordering methods. The implementation is available as an R package, named “dendsort”, from the CRAN package repository. The R script for generating figures in this paper is available as a supplementary material. Further examples, documentations, and the source code are available at [https://bitbucket.org/biovizleuven/dendsort/].

6.6.3 Methods

Hierarchical clustering

Agglomerative hierarchical clustering (HC) starts with individual observations as singleton clusters and merges clusters iteratively until all clusters belong to one big cluster. In each iteration, the two most similar clusters are identified by a distance measure and a linkage algorithm of choice. The details of the algorithm and the properties of distance measures and linkage algorithms are described in [119, 120, 127].

The default hierarchical clustering method in R combines three types of merges: a merge between two singleton clusters, a merge between a singleton cluster

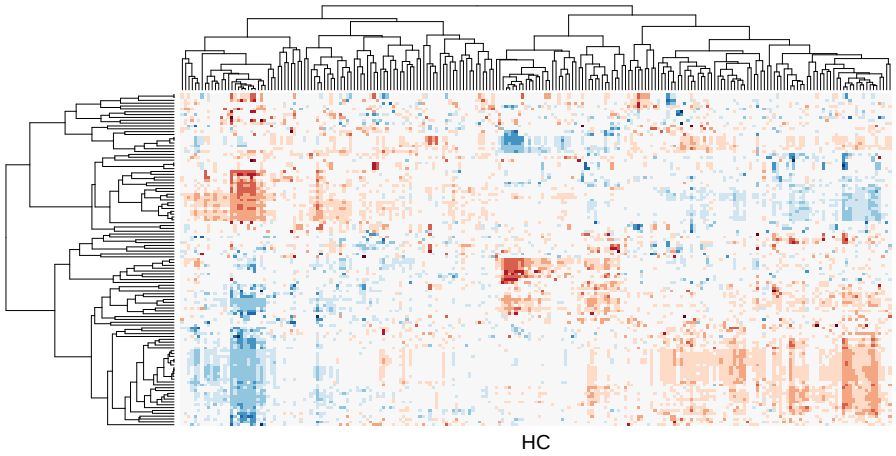

Figure 6.15: Cluster heat map of the data matrix from the integrated pathway analysis of gastric cancer from the Cancer Genome Atlas (TCGA) study.

and a cluster with more than one member, and a merge between clusters with multiple members. The heuristics for determining the orientation of merging elements essentially determine the structure of the resulting dendrogram.

Using a simple two-dimensional data set as shown in Figure 6.16A, we demonstrate the default heuristics used in the hierarchical clustering method in R. A dendrogram is constructed as follows: When a leaf (singleton cluster) merges with another leaf, the orientation of clusters is determined by the order of observations in the input data matrix, as seen in branch “a”, “b”, “c” and “f” in Figure 6.16B. When a leaf merges with a cluster with more than one member (subtree), the leaf is always placed on the left side of the branch, as shown in branch “d” and “g”. When two subtree merges, the subtree with the smaller distance in the previous merge is placed on the left, as seen in branch “e”, “h”, and “i”. Each branch is labeled alphabetically in the order of merges within the clustering process.

In contrast to the default heuristics, our heuristics are characterized by 2 key differences: first, a leaf is placed on the right side when it merges with a subtree; second, when two subtrees merge, the subtree with the smallest distance among all of preceding merges is placed on the left (Figure 6.16C). The first rule avoids a branch of a singleton cluster hanging over the preceding nested clusters and allows the tree to grow from left to right in the order of merges. The second rule ensures that the tightest cluster is placed leftmost within the subtree structure. Consequently, our heuristics result in each subtree or sub-cluster structure in a right triangular shape, as shown in Figure 6.16C. This feature increases the

contrast between the items at the edge of adjacent subtree structures, thus modularizing each subtree structure.

The MOLO method takes the result of the default hierarchical clustering method, and re-evaluates the orientation of the clusters at each branch recursively. The pseudocode of this algorithm is shown in Figure 6.17. In addition to the algorithm based on the smallest distance, we also implemented a variant in which the average distances of all preceding merges are compared, and discussed further in the third case study. The data in Figure 6.16 consist of only 10 observations and it is merely intended to explain the difference in heuristics. The following case studies demonstrate applications of the MOLO algorithm with larger datasets, and compare visualizations created using our heuristics and other existing leaf ordering methods.

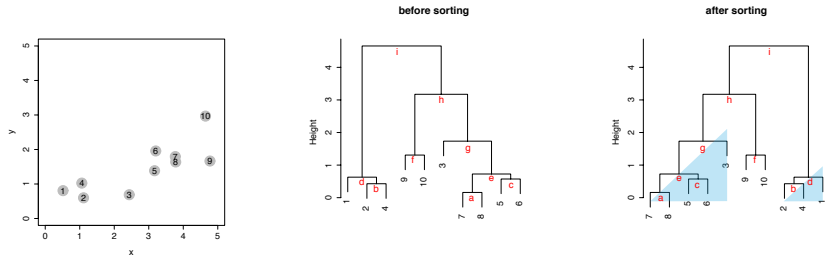

Figure 6.16: Hierarchical clustering of a simulated two-dimensional data set. (A) A scatterplot of the ten input elements. The number of each element also represents the order in the input matrix. (B) A dendrogram drawn using the default heuristics in R. The branches in the dendrogram are labeled from "a" to "i" in the order in which clusters are merged. (C) A dendrogram reordered using MOLO with the smallest distance. The global structures in a shape of the right triangle are higlighted.

6.6.4 Results

Case study 1: Comparison of clustering algorithms

One of the key tasks in applying hierarchical clustering is to choose an appropriate distance metric and a linkage algorithm [127]. A choice of distance metric, such as Euclidean distance and correlation-based distance, defines a measure of similarity between two elements. Clustering algorithms, such as complete, average, and single linkage, are variations of the cluster proximity definition [120]. The choice of distance measures and linkage algorithms influences the clustering results. It is therefore recommended to try different

```
sort_smallest(d){
  //d is a dendrogram object which consists of
  //nested dendrogram objects on its left and right,
  //d_l and d_r.
  if d_l and d_r are singleton clusters
    add the minimum distance to d
    return d
  else if d_l is a subtree and d_r is a singleton cluster
    sort_smallest(d_l)
    set d_l to the left and d_r to the right side of d
    add the minimum distance to d
    return d
  else if d_l is a singleton cluster and d_r is a subtree
    sort_smallest(d_r)
    set d_r to the left and d_l to the right side of d
    add the minimum distance to d
    return d
  else if d_l and d_r are subtrees
    sort_smallest(d_l)
    sort_smallest(d_r)
    if the minimum distance of d_l < the minimum
    distance of d_r
      set d_l to the left and d_r to the right side of d
    else
      set d_r to the left and d_l to the right side of d
    end if
    add the minimum distance to d
    return d
  end if
return d
```

Figure 6.17: The recursive algorithm for ordering a dendrogram structure based on the minimum distance.

HC settings in exploratory data analysis, especially when the underlying data structure is unknown.

As Hastie et al. [119] point out, dendrogram structures can vary greatly depending on the choice of linkage algorithms. In Figure 6.18, dendrograms of different linkage algorithms for the same simulated data set are compared. The appearance of the dendrogram structure is quite different and it is difficult to compare similarities in the nested cluster structure. In contrast, when the MOLO method is applied, we find the reordered dendrograms easier to study the nested structure and to compare between one another (Figure 6.19), because

the linear leaf order in these dendrograms reflect the order in which clusters are merged. For instance, the element 32 and 34 form the tightest cluster, and they are easy to identify because they are always placed leftmost. Also, upon closer examination of the reordered dendrogram structures, we find that the reordered dendrograms reflect the underlying difference in algorithms more closely. For example, the average linkage is an intermediate approach between the single and complete linkage algorithms to define cluster proximity [120]. Although the MOLO method does not change the clustering result itself, this case study demonstrates how it can improve, or at least bring a new perspective, to interpret dendrogram structures.

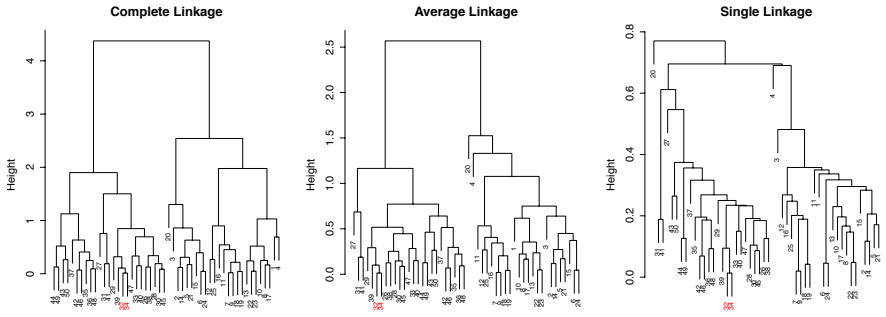

Figure 6.18: Comparison of dendrograms from different linkage algorithms using R's default ordering heuristics. The element 32 and 34 are highlighted.

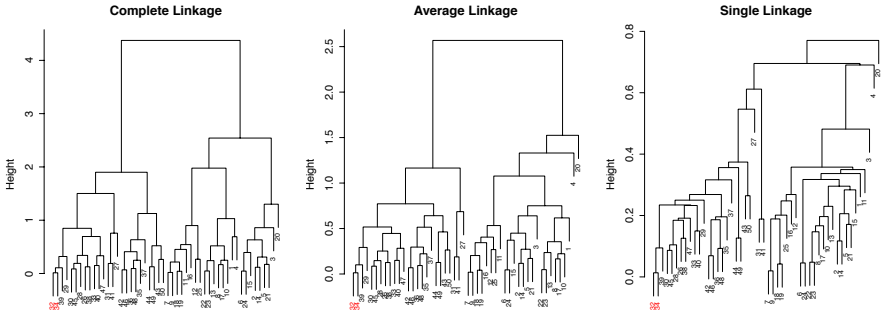

Figure 6.19: Comparison of dendrograms from different linkage algorithms after applying the MOLO method based on the smallest distance. The element 32 and 34 are highlighted.

Case study 2: Iris data

The second case study extends the demonstration of seriation-based leaf ordering methods by Buchta et al.[128] using Fisher's Iris data set. Fisher's Iris data set is available from the R's *dataset* package [129].This Iris data set represents 3 species of iris with 50 observations for each species. Each observation contains measurements of 4 attributes: the sepal length and width, and the petal length and width. In this case we performed hierarchical clustering on the distance matrix of Euclidean distances, using the complete linkage algorithm. In Figure 6.20, adjacency matrices are visualized as cluster heat maps to compare results of the default hierarchical clustering (HC), the Gruvaeus and Wainer's method (GW)[124], the optimal leaf ordering (OLO)[121], and the MOLO method (MOLO). These matrices are diagonally symmetric and rows and columns are reordered based on the leaf order of dendrograms. The species for each observation is color coded and shown between the dendrogram and the heat map visualization. Implementations of the GW and OLO methods are available in the *seriation* R package[128].

Despite the fact that each representation shares the same underlying hierarchical clustering output, the visual impressions of heat maps are different depending on the choice of leaf ordering methods. For example, the results of the HC, GW, and OLO methods suggest two predominant clusters, as indicated by dark square blocks along the diagonal axis. On the other hand, the result of the MOLO method suggests three clusters. The MOLO heuristics place the most similar items on the left ends of each subtree structure and subsequently merged clusters are placed on its right. As a result, the MOLO method reorders the dendrogram structure to reflect the modularity of the cluster-subcluster structure. With the information of species for each observation, it becomes clear that there are three species and a half of *versicolor* samples are clustered together with *virginica*.

Additionally, we find the cluster edges in the heat map visualization of the MOLO method are more prominent than those of other leaf ordering methods. One explanation for the enhanced edges is the increased contrast between subtree structures, whereas the GW and OLO methods aim to reduce the edge contrast between subtree structures, resulting in more fuzzy boundaries. This effect can be seen at the borders between *versicolor* and *virginica* species in heat map visualizations. The second explanation is that the monotonic linear order results in an optical illusion, called Mach band effect, at the edge of subtree structures. The Mach band effect explains how edges in different shades of gray have exaggerated contrast when in contact [15]. This enhanced edge-detection works to our advantage in identifying clusters, especially because our visual systems to decode quantitative or continuous data from different shades of

colors is limited [130].

As also pointed out in previous studies [121], the GW and OLO methods result in a global structure where highly similar items appear in the middle, while marginally related items are on the edge of the subtree structure. This tendency is most apparent in the *setosa* samples. On the other hand, the MOLO method results in a right triangular global shape where the similarity of clusters increases from left to right, unidirectionally, for each subtree structure. This global property enhances the contrast at the borders of clusters and reveals the third cluster in the heatmap visualizations.

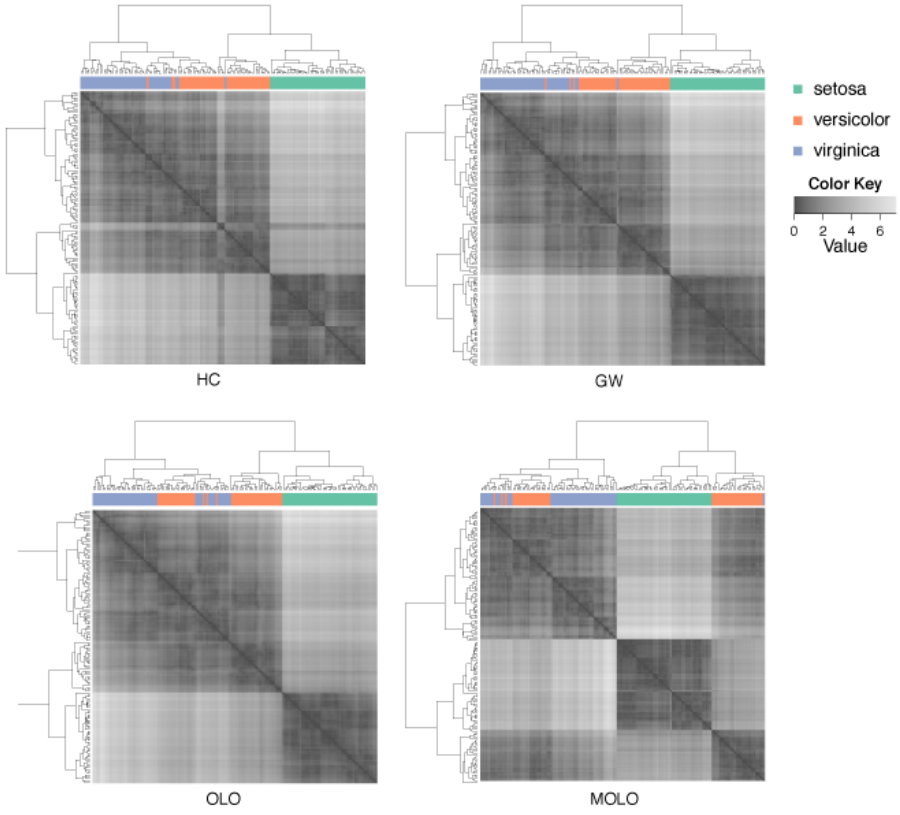

Figure 6.20: Comparison of leaf ordering methods in cluster heat maps. The default hierarchical clustering (HC), the Gruvaeus and Wainer's method (GW), the optimal leaf ordering (OLO), and the MOLO method are applied to the Fisher's Iris data set.

Case study 3: TCGA

The third case study involves a multivariate table obtained from the integrated pathway analysis of gastric cancer from the Cancer Genome Atlas (TCGA) study [106]. In this data set, each column represents a pathway consisting of a set of genes and each row represents a cohort of samples based on specific clinical or genetic features. For each pair of a pathway and a feature, a continuous value of between 1 and -1 is assigned to score positive or negative association, respectively. The goal of this cluster analysis is to explore patterns in the data set and examine clusters to characterize the link between the gene expression levels, and clinical features and to identify subtypes of the cancer among the cohort of samples.

These matrices are typically visualized as cluster heat maps (Figure 6.15). By applying hierarchical clustering on the rows and columns independently, the rows and columns are reordered to place similar items close to each other. In this example, the distance measure is based on the Pearson's Correlation coefficient and the complete linkage algorithm is used for hierarchical clustering.

Similarly to previous examples, the application of the MOLO method results in a global right triangular shape for each subtree, encoding the monotonicity of the hierarchical clustering process (Figure 6.21). However, upon a closer examination, we find that the first subtree of the rows does not form a right triangular shape. This first cluster is a very loose cluster having relatively long branches, except for the very first two rows which have the shortest distance. The characteristic of a loose cluster is also reflected in the heat map visualization, where there are no strong patterns of clustering, except for the first two rows. In order to prioritize tighter clusters with a smaller average distance, we implemented a variation of the modular leaf ordering method based on the average distance of the preceding merges (MOLO_AVG). The effects of leaf ordering methods on dendrogram structures for the rows are compared in Figure 6.22. With the MOLO_AVG method, the tight clusters with lower average distances are placed leftmost.

The cluster heat map generated with the MOLO_AVG method is shown in Figure 6.23. The choice of either the smallest or average distance does not influence the structure within subtrees, however the order of the subtree structures changes. Although the difference may be subtle, we find that the modularity of clusters becomes more distinctive with the MOLO_AVG method. The resulting visualization also provides new insights into relationships between clusters. For instance, the inverse relationship between sets of rows and columns becomes more apparent in Figure 6.23 than the original figure (Figure 6.20).

One way to evaluate the efficiency of a graphical representation is to compare the

proportion of ink used to represent the data, a concept known as the data-ink ratio [131]. Since each dendrogram shares the same underlying hierarchical clustering output, the total length of lines required to draw a dendrogram can be directly compared to evaluate the conciseness of dendrogram representations. We calculated the total length of lines used to draw dendrograms in Figure 9, the results of which are shown in Table 6.1.

The MOLO_AVG method results in the highest reduction in the data-ink ratio, while the GW method results in an increase in the data-ink ratio. Since the total number of vertical lines in each dendrogram is the same, the difference in the total length is due to the horizontal lines. A factor contributing to the reduction of horizontal lines is the heuristic of placing the singleton cluster on the right side of the branch. This heuristic avoids the placement of a singleton cluster on the left side, spreading over the nested tree structure.

As the data size increases, the number of rows or columns in the data matrix increases while the display space for the figure may be limited. As a result, a dendrogram representation may become denser with more leaves, making the details of hierarchical structure harder to read. Figure 6.24 shows the same dendrograms as in Figure 6.22, but in a more limited display space. Because the MOLO methods results in a global pattern of right triangular shapes, it supports the viewer to identify tight and loose clusters even when the vertical lines of branches are so dense that they are in contact with adjacent branches. Similarly, because of this right triangular shape, each subtree structure is still distinguishable. Therefore, the MOLO methods aid the readability of dendrogram structures, even when the display size is limited.

In summary, this case study demonstrates how the MOLO methods support tasks in exploratory data analysis and improve readability of the dendrogram representations by reducing visual clutter. The dendrogram structure after the MOLO methods results in right triangular shapes for each subtree structure, and the order of leaves in each subtree reflects the order in which clusters are merged. In common with the case study of the Iris data set, the MOLO methods aid cluster identification in cluster heat maps.

Method	HC	GW	OLO	MOLO	MOLO_AVG
Total length	559.79	598.93	551.98	492.88	437.48
Ratio to HC	1	1.07	0.99	0.88	0.78

Table 6.1: Comparison of the total line lengths required to draw the dendrogram structures shown in Figure 6.22.

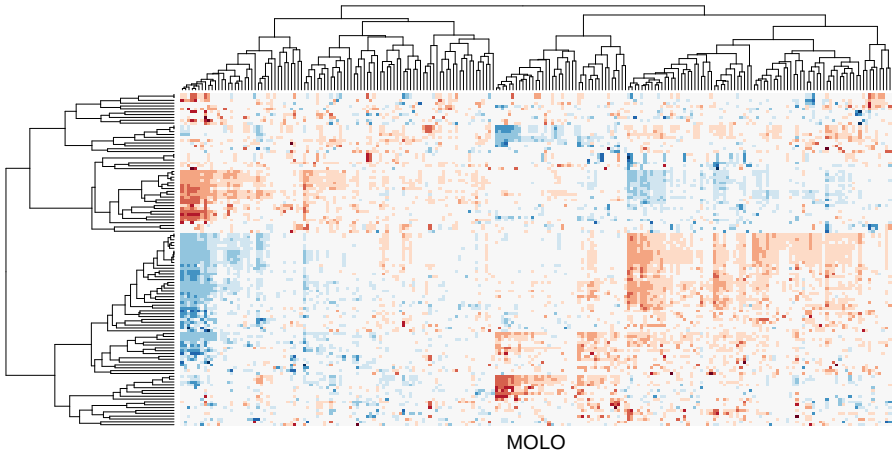

Figure 6.21: Cluster heat map of the data matrix after applying the MOLO method based on the smallest distance.

6.6.5 Discussion

In this paper, we introduce two modular leaf ordering methods and demonstrate how leaf ordering of dendrograms can influence the interpretation of cluster heat map visualizations. While seriation-based leaf ordering methods focus on homogenizing the linear order of leaves, our heuristics focus on improving the graphical representation of dendrograms to reflect the intrinsic properties of the hierarchical clustering process, such as the monotonic increase of distances in successive merges. As a result, each subtree structure has a global right triangular shape. This modular property is also reflected in the linear order of leaves, thus influencing the visual impression of clusters in heat map visualizations.

Although the leaf ordering methods affect the dendrogram representation and the linear order of leaves, it does not change the underlying hierarchical structure. In other words, the quality of the clustering results ultimately depends on the quality of the input data and the choice of appropriate distance metric and linkage algorithm. Given no prior knowledge of underlying patterns in data sets, it is recommended to try different normalization techniques in preprocessing and different distance measures and linkage algorithms to allow different aspects of the data to be explored [127].

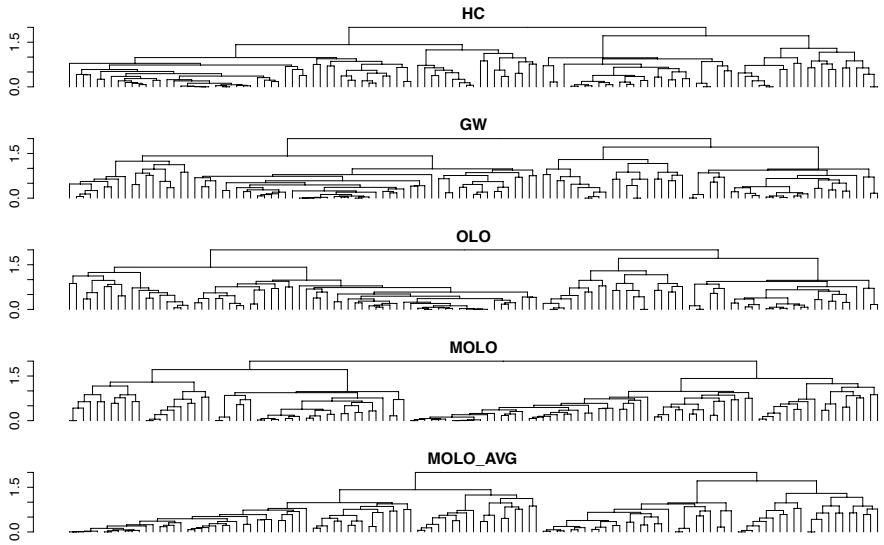

Figure 6.22: Comparison of dendrogram structures resulting from different leaf ordering methods. The rows from the example data sets are shown.

6.6.6 Conclusions

Through case studies, we demonstrate the effects of our leaf ordering methods on the interpretation of the clustering result, as well as the reduction in visual clutter as measured by the data-ink ratio. With cluster heat map techniques being very popular in life sciences, we advocate our methods to be considered both for exploratory data analysis and for publication of figures.

6.6.7 Software Availability

Software access

http://cran.r-project.org/web/packages/dendsort/index.html

Latest source code

https://bitbucket.org/biovizleuven/dendsort/

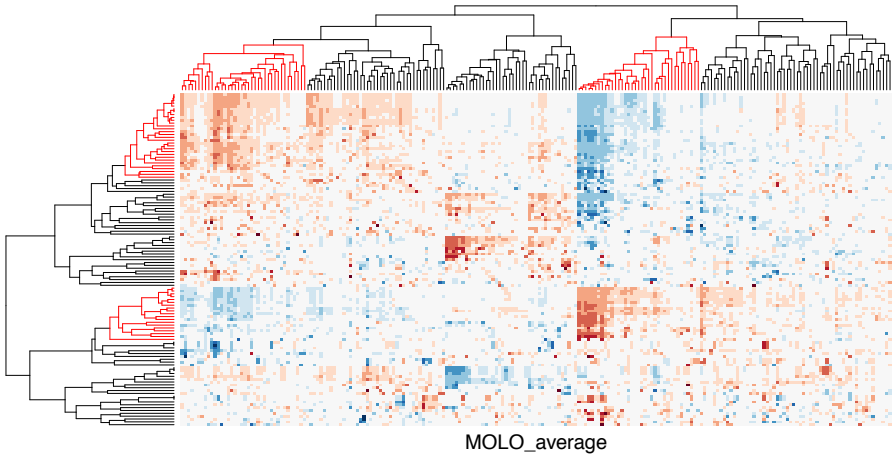

Figure 6.23: Cluster heat map of the data matrix after applying the MOLO method based on the average distance. The rows and columns with an inverse relationship are highlighted in the dendrograms.

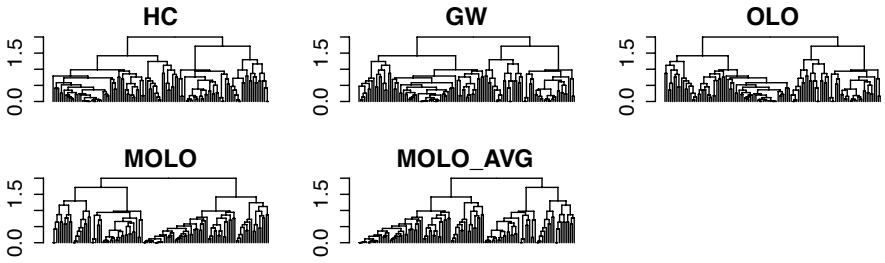

Figure 6.24: Comparison of dendrogram structures resulting from different leaf ordering methods in a limited display space. The rows from the example data sets are shown.

Source code as at the time of publication

https://bitbucket.org/F1000Research/dendsortarchive

Archived source code as at the time of publication

http://dx.doi.org/10.5281/zenodo.10980

Software license

GPL-2 | GPL-3

Author contributions

All authors contributed to the design and organization of the paper and its writing and editing. RS initiated the project. TV and RS implemented the methods in R. RW and AVdM were involved in discussions for the development and JA supervised the project.

Competing interests

No competing interests were disclosed.

Grant information

This work was performed under the umbrella of the KU Leuven Data Visualization Lab (www.datavislab.org) and supported through funding from the KU Leuven Research Council CoE PFV/10/016 SymBioSys (RS), the Academische Stichting Leuven vzw (RS), the IWT O&O ExaScience Life Pharma (TV), and iMinds ICON b-SLIM (RW).

Acknowledgements

We would like to thank Sheila Reynolds and Vésteinn Þórsson from the Institute for Systems Biology for sharing the sample data set for the third case study.

Chapter 7

Data Acquisition and Transformation

7.1 Introduction

With advances in biological data acquisition technologies and computational analytic methods, the integrative analysis approach has become mainstream where new datasets are combined, and various computational approaches are applied to refine datasets iteratively. Consequently, the visualization techniques and conventions established prior to the advent of new technologies need to adapt and evolve together with the data to support the integration of multiple datasets and to meet biological analysis needs. Figure 7.1 highlights the **Acquisition & Transformation** process, where the input is based on the domain knowledge and tasks and output is the data used for visualization. *Acquisition & Transformation* involves an acquisition of new datasets through new experiments, integration of existing data, and transformation of the data.

In this chapter, two design studies are discussed: Oligoprobe visualization and Singular Value Decomposition (SVD). The Oligoprobe project integrated existing datasets, such as Gene Ontology terms and Kegg pathway annotations, as well as the outputs of Self Organising Maps (SOMs) to cluster transcripts based on the temporal expression patterns. The second project is an extension of scatter plot matrix (SPLOM) and biplots. It addresses an aspect of the output that are often overlooked in, for example, a closely related factorisation method such as Principal Component Analysis (PCA). These case studies highlight the role of *Acquisition & Transformation* in our vis design framework (Figure 7.1).

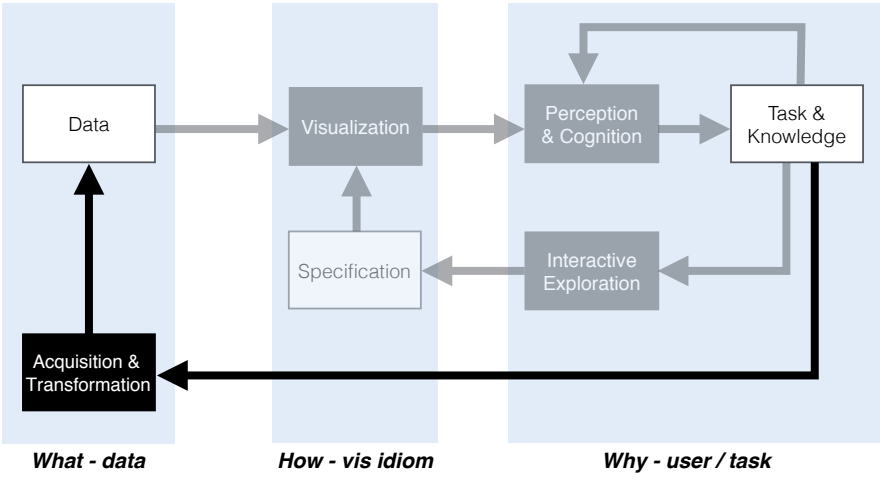

Figure 7.1: A framework for vis design, highlighting Acquisition & Transformation.

7.2 Case Study: Oligoprobe

Collaboration:

Ligia Mateiu[1], Dusan Popovic[2] and Thierry Voet[1]

L.M. and T.V. were domain experts and target users. D.P. performed the Self Organising Map and analyzesed output data.

[1]*KU Leuven, Department of Human Genetics, Leuven, Belgium*
[2]*KU Leuven, ESAT - STADIUS, Leuven, Belgium*

The oligoprobe design study aimed to support research involving characterisation of zygotic and early embryonic transcripts, especially the function of shorter 3' Untranslated Regions (UTRs) and the process of maternal transcript degradation. The domain expert developed a bioinformatics pipeline to detect transcripts and their 3'UTR lengths from messenger RNA (mRNA) expression profile data. The domain expert visualized the probe-level expression for each transcript as a heatmap as shown in Figure 7.2. The probes were ordered along the y-axis based on their genomic positions and the samples from different development stages were shown along the x-axis. Then, the lengths of the transcripts were inferred from the expression levels across these stages. The analysis challenge at the beginning of collaboration was that the output of data processing had two to three thousand unique transcripts and analysing each transcript as a heatmap was tedious and overwhelming both perceptually and cognitively.

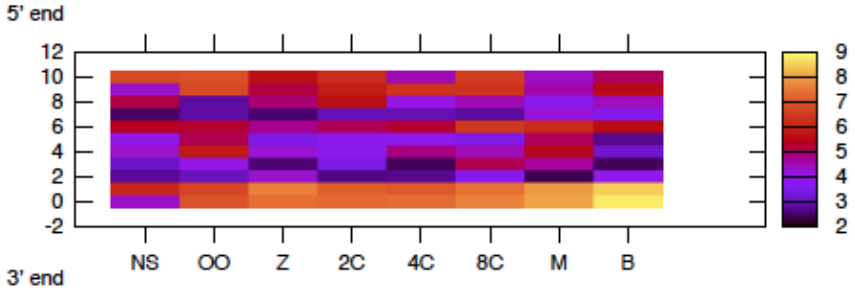

Figure 7.2: A heatmap visualization of a transcript expression profile measured at eight different time points.

The heatmap visualization technique is a very space efficient method to represent a tabular dataset, however, the use of colour to encode quantitative data comes

with its limitations. Although the heatmap gives a good overview of the data set, it is not intuitive when you try to compare values across a row or column because we do not perceive an intrinsic order in the hue. Using saturation or brightness would result in a more accurate visual encoding of quantitative values, but would not be as accurate as the position or the length visual channels [16, 7]. This limitation of the use of colour to encode quantitative value is also raised by other visualization researchers [109, 107]. This limitation was our initial motivation for the project, and a starting point of this collaboration was to find an alternative visual encoding.

The first prototype visualised the same data using the parallel coordinates vis idiom (Figure 7.3). In this visual encoding, each vertical axes represents a development stage, and each polyline represents an expression profile of a probe. The polylines are colour coded based on the position of the probe on the transcript. The coding region is coloured blue, and the UTR is coloured magenta. Because this vis idiom uses the position to encode the quantitative value, the range of perceptually discriminable values is much wider than that of the hue. The analyst found it much easier to discern what was the noise and the pattern in this view. This prototype also included a simple interactive function to filter based on the average trends between two developmental stages. Also, genomic coordinates of the probes along its transcript were visualised on the right-most axis.

The subsequent iteration combined a linked scatter plot to provide an overview of all the transcripts for data exploration (Figure 7.4). The scatter plot shows the difference in average values between two stages. For example, the scatter plot in the figure shows the value difference between the spermatozoa and oocyte stages for the UTR and the coding region. The profile of a selected transcript is shown in the parallel coordinates on the right. This prototype allows the user to explore all transcripts by examining a specific interval at a time. However, it was still laborious to go through thousands of transcripts at different development stages and difficult to glean any unique recurring patterns.

After consulting with a machine learning expert, we decided to use a SOM to cluster transcripts based on the average expression profile of the probes in the UTR and the coding region. A SOM consists of nodes arranged on a two-dimensional grid. A rectangular grid was used to "train" the model. The output of a SOM was a grid of nodes, and each node represented a cluster of transcripts. The output grid preserved the topological properties of the input space. This grid layout is typically visualised by encoding the distance between two adjacent nodes, as shown in Figure 7.5. In this visual encoding, each square representing a node is divided into quarters. Each quarter is then coloured based on its distance to its adjacent node. However, this visual encoding does not show how many transcripts are assigned to each node. Thus, a new visual

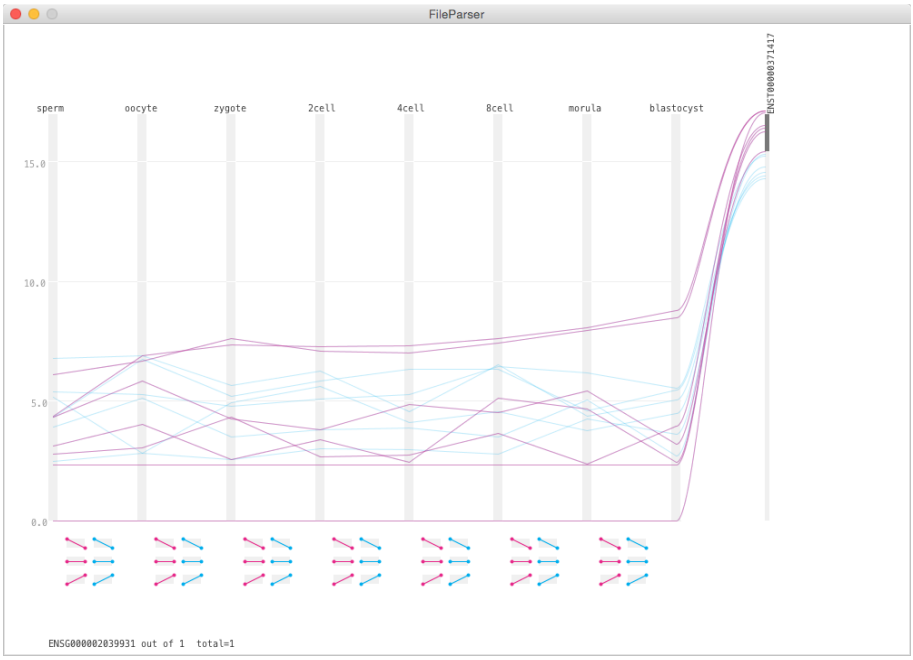

Figure 7.3: The first prototype using the parallel coordinates visual encodings. The probes are colour coded based on their position on the transcript. The probes in the coding region are in blue and probes in the untranslated region are in magenta.

encoding was devised (Figure 7.6). The line thickness showed the average distance between nodes. The area of a circle for each node represented the number of transcripts assigned to the node. We found this new visual encoding more intuitive to interpret the topology of clusters. This visualization prototype helped to gain a better understanding of the SOM performance and its outputs, and the grid was relatively large (30x30 nodes).

Encouraged by the result of the preceding prototype, the next iteration of prototype integrated the SOM result to characterise clusters and to investigate their biological functions. For this iteration, the SOM was trained using a smaller rectangular grid (10x10). The SOM result is shown on the bottom left of the interface (Figure 7.7). By selecting a node in the SOM representation, the subset of corresponding transcripts is shown on the scatter plot. When a transcript is selected from the scatter plot, the details of the mRNA expression profile are shown in the parallel coordinates view. The thicker lines in parallel coordinates showed the median or mean, depending on the setting. The smaller parallel coordinates plot underneath showed the average profile of the cluster, which

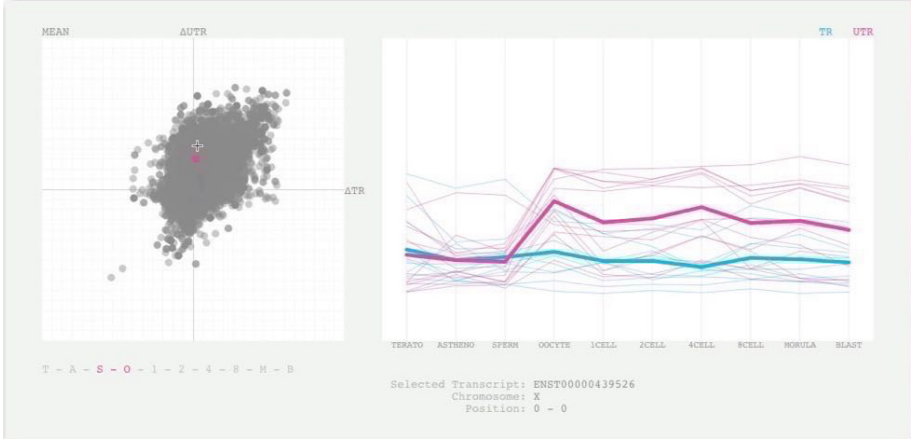

Figure 7.4: The second prototype with a scatter plot and a parallel coordinates plot. The scatter plot shows the difference in average values at two stages.

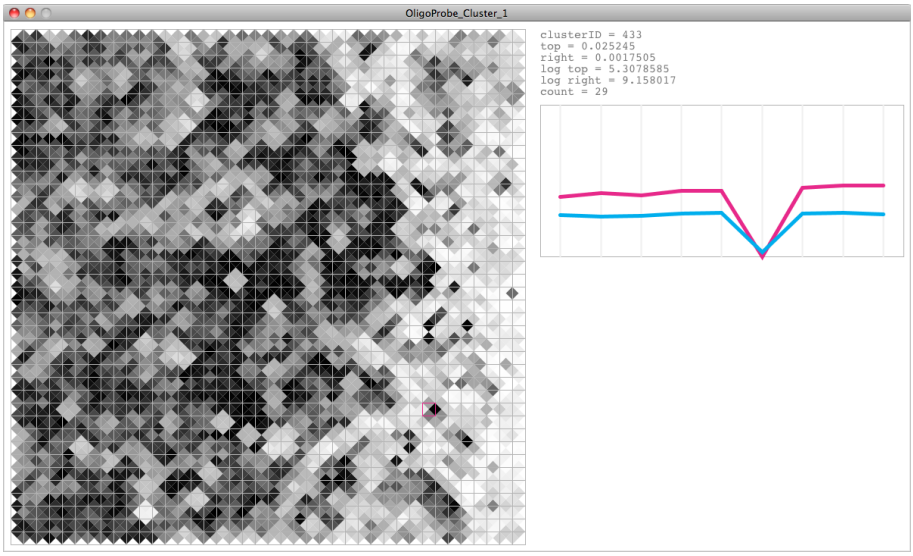

Figure 7.5: Visualization of a SOM output.

also known as a "prototype". (Visualization prototype and cluster prototype as a result of the SOM should not be confused.) The textual information in the middle provided the information about the size of the cluster, the gene ontology enrichment, and the external links to the Ensembl genome browser and the

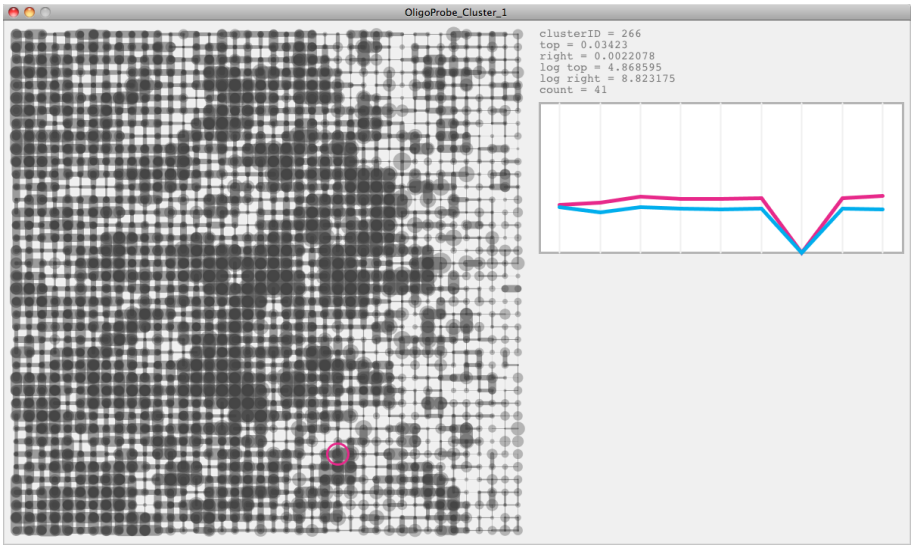

Figure 7.6: Redesigned visual encoding for a SOM output.

Gene Cards. During this design iteration, the domain expert went back and forth between this vis tool and revising data preprocessing methods. The vis tool provided the analyst some useful insights into the noise in the data and how the preprocessing can be improved.

The vis tool included alternative visual representations for the multi-panel view. As an alternative to the parallel coordinates plot, the heatmap representation was reintroduced at the analyst's request (Figure 7.8). Figure 7.8 encodes the same data as shown in Figure 7.7. Although parallel coordinates plots are perceptually more accurate than the heatmap in encoding the quantitative data, the analyst appreciated having the option to view the data in the way they were used to. A lesson for vis designers here is to take a slow transitional process to introduce a new visual encoding unfamiliar to the user [35] and to take user's previous experience into account. It also had a small multiple view for the average profile of the SOM output (Figure 7.9). The SOM output is a grid of nodes, and the grid preserves the topology of input data. Hence, the small multiples of the average profile provided an intuitive interface for the user to select the cluster of interest and to explore its related neighbour clusters.

The last iteration changed the angle of attack from the exploration of clusters to the exploitation of the domain knowledge to identify the cluster of interests. During the evaluation of the prototype, the domain expert expressed a wish to explore the data using the known and existing functional annotation of the

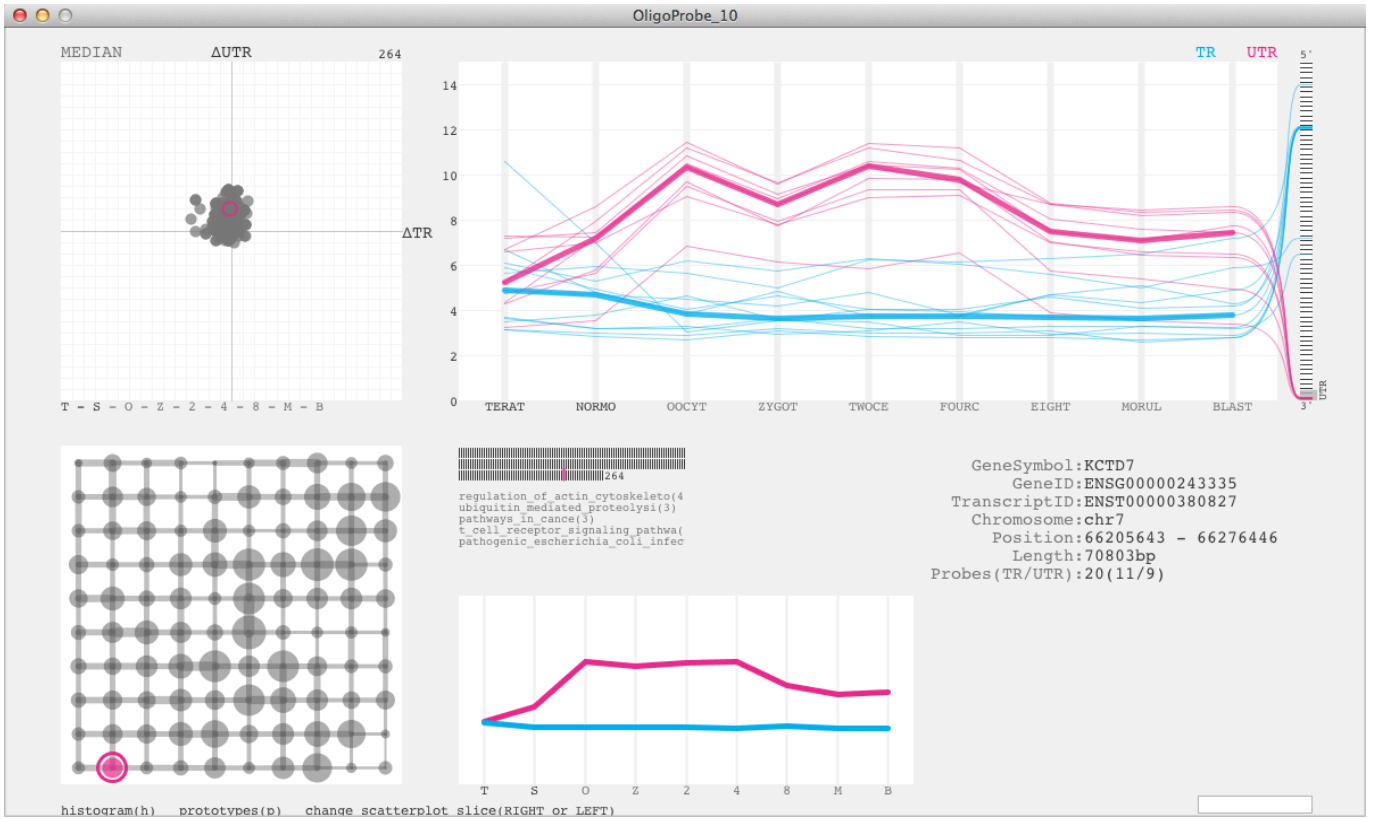

Figure 7.7: The interface of a vis prototype integrating the SOM outputs and functional annotations of transcripts.

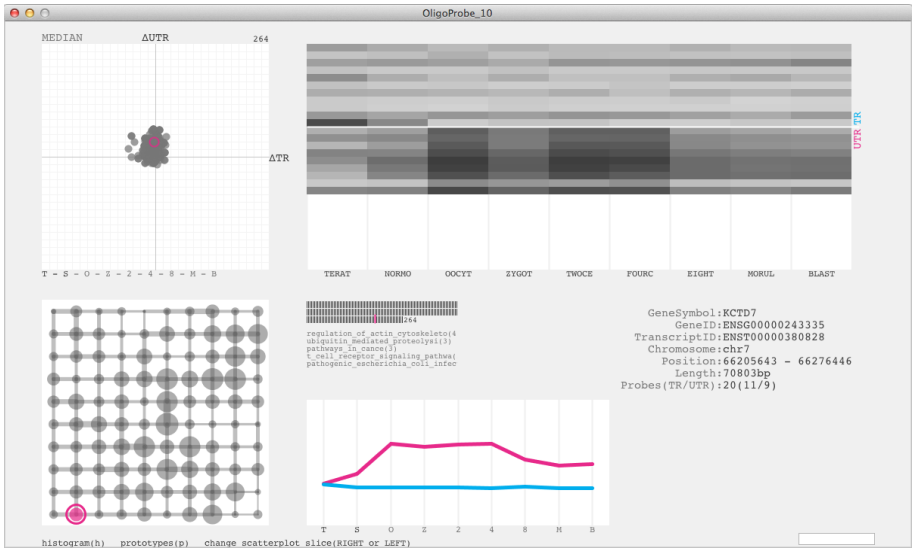

Figure 7.8: The heatmap mode as an alternative to the parallel coordinates plot.

genes. For example, the domain expert was specifically interested in those genes that are involved in the spermatogenesis. To enable the search of clusters based on the functional annotation of genes, the new prototype integrated the functional annotation from the Kegg pathway or Gene Ontology (GO) terms (Figure 7.10). The interactive list of terms with a text search function was provided on the left. A selection of a term from the list showed clusters that had the highest count of genes annotated with the selected term. The main objective of this view was to leverage the domain specific knowledge to search for clusters of interest in conjunction with the unsupervised learning outputs.

Even though this design study didn't lead to a publication, the design process included dynamic interactions between the domain expert and the vis designer. Each iterative prototype provided new insights into the data and refined research questions and analysis tasks. The prototypes combined both visual analytics and computational approaches. This design study highlights the role of *Acquisition & Transformation* in the vis design framework (Figure 7.1). The first few iterations led to apply a SOM to transform the data by clustering transcripts. Then, the interactive prototype provided useful insights to the domain expert to revisit and improve the data preprocessing pipeline. The unsupervised learning approach, such as a SOM, together with visual representations of its output empowers the domain expert to pursue questions and uncover both expected and unexpected patterns in exploratory data analysis.

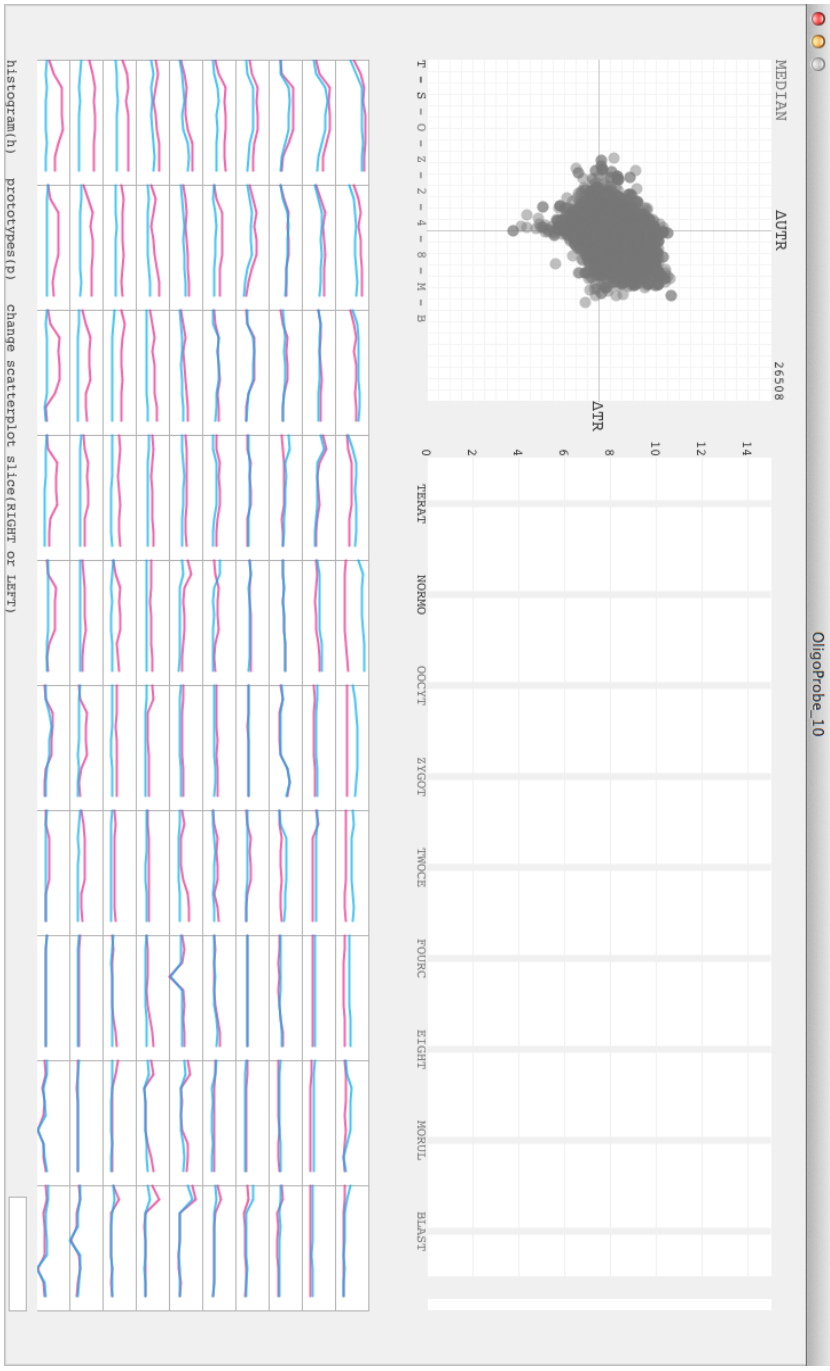

Figure 7.9: The small multiples of the average profiles of the SOM output.

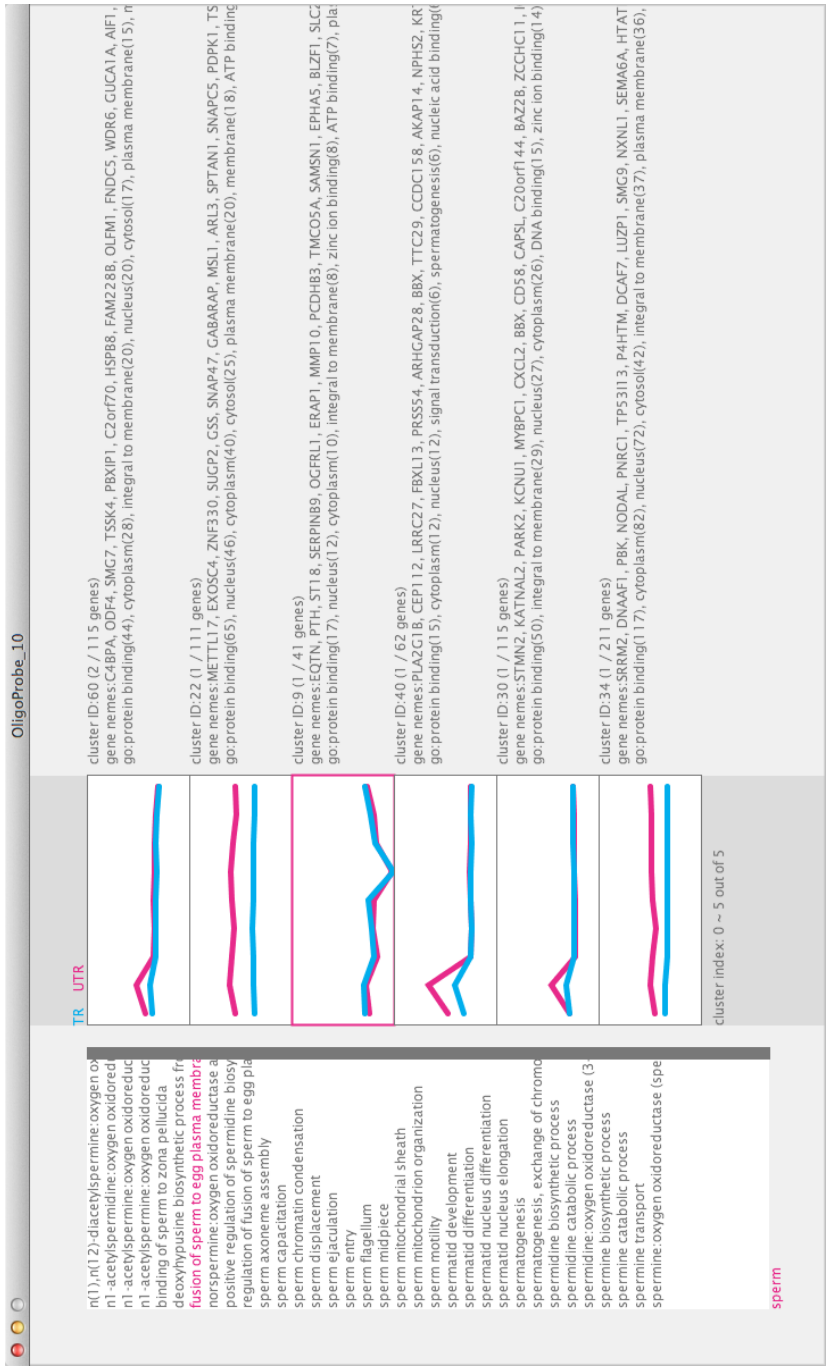

Figure 7.10: The analysis of the SOM outputs using the functional annotation of genes.

7.3 Case Study: Biplot Matrix

Collaboration:

Jaak Simm[1], Adam Arany[1], Yves Moreau[1] and Tim Xiaoming Hu[2]

J.S., A.A. and Y.M. performed data analysis on the drugs and compounds and provided the input data for the prototype. T.H. was the domain expert and the target user for the single cell data.

[1]*KU Leuven, ESAT - STADIUS, Leuven, Belgium*
[2]*European Molecular Biology Laboratory - European Bioinformatics Institute (EMBL-EBI), Wellcome Trust Genome Campus, Hinxton, Cambridge, UK*

The second design study is concerned with visualizing the outputs of the Singular Value Decomposition (SVD). SVD is a factorisation method with a wide range of applications in signal processing and statistics. SVD is closely related to PCA, which is a common unsupervised approach in exploratory data analysis in biology. The motivation of this design study was to improve the interpretation of the output matrices in biological contexts.

Using SVD, an arbitrary matrix $M(n \times m)$ can be decomposed to an orthogonal matrix $U(m \times r)$, a diagonal matrix $\Sigma(r \times r)$ and another orthogonal matrix $V^T(n \times r)$ [132]. The columns of U are the left singular vectors, and the rows of V^T are the right singular vectors of the matrix M. The diagonal values of matrix Σ are the singular values of the matrix M. Singular values are listed in descending order. One application of SVD is the low-rank matrix approximation, where an approximation of the matrix M is derived based on a specific rank r. For example, a table of simulated gene expressions of samples is decomposed, and the approximation of the input matrix can be made using only r singular values and vectors (Figure 7.11). To further examine the relationship between the left and right singular vectors, we first extend the scatter plot matrix vis idiom (Figure 7.12). Whereas Figure 7.12 shows the first five single values, Figure 7.13 highlights the first three single values by colouring based on the known gene sets and sample cohorts as shown in Figure 7.11.

In addition, we extended the biplot vis idiom via the linking-and-brushing interaction [133]. Biplot overlays scatterplots showing information of both U and V orthogonal matrices. For example, when data points in the U singular values are selected, those data points are represented as arrows in the scatter plots of V singular vectors as $U\Sigma$ (*raw principal component scores*). Similarly, when data points in the V are selected, those data points are represented as arrows in the scatter plots of U singular vectors as $V\Sigma$ (*variable-component*

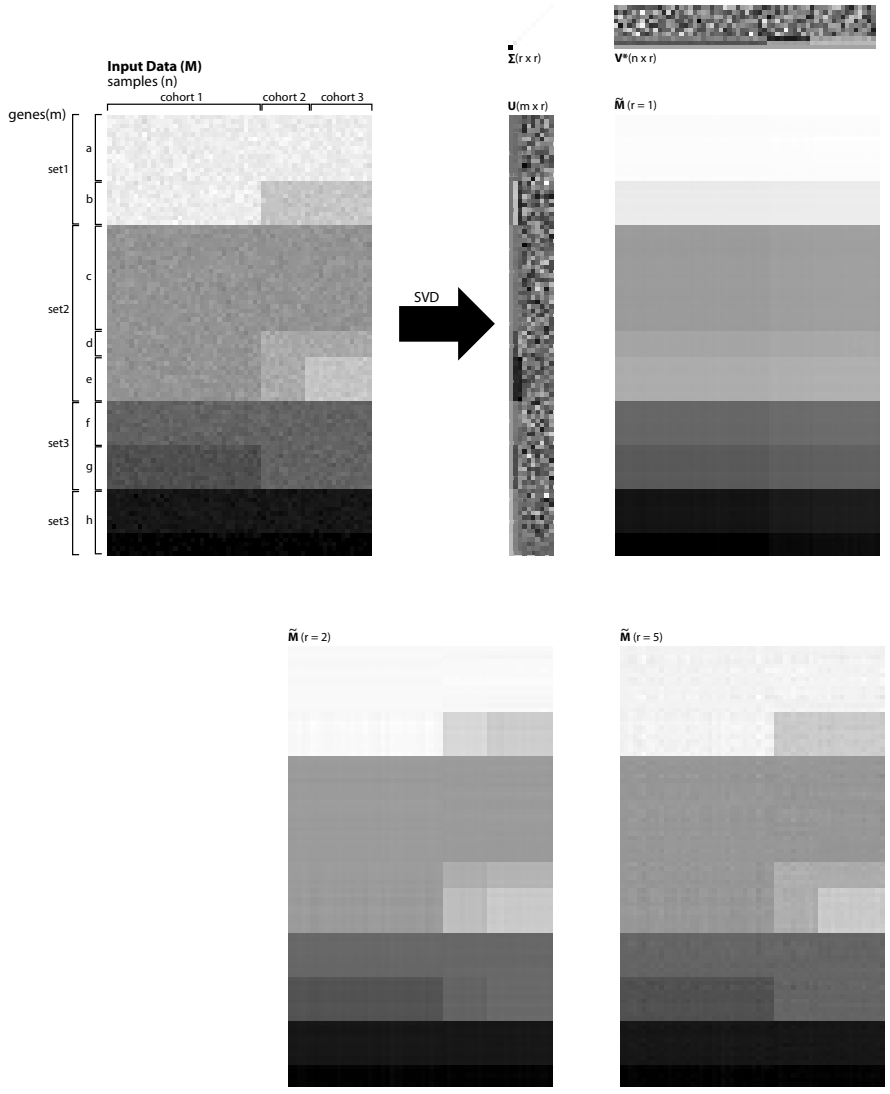

Figure 7.11: Singular value decomposition of simulated gene expressions and its low-rank matrix approximations for $r = 1$, $r = 2$ and $r = 5$.

loadings). The reason for the use of interaction is to reduce the visual clutter since the input matrix M tends to be large and the scatter plots themselves may be already quite dense. We call the combination of biplot and SPLOM as

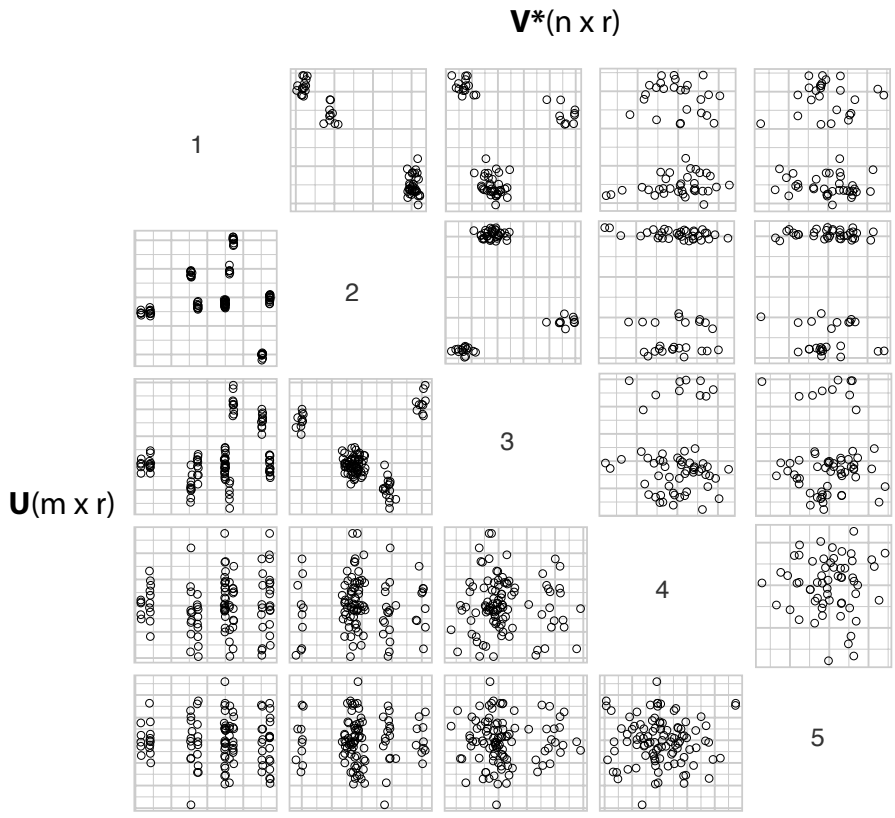

Figure 7.12: Scatter plot matrix of singular vectors (r=5).

biplot matrix.

The *biplot matrix* vis idiom was evaluated with two datasets: a matrix of drug targets and compounds and a matrix of gene expressions and single cell samples. The drug target and compound dataset included extra information on known interactions between compound and targets. The vis tool included a function to retrieve the compound structure and a filter function based on the $P^{IC}50$ score (Figure 7.14). The compounds were coloured in red, and the targets were coloured in blue. For the gene expression dataset, the number of genes was much higher than the number of samples (Figure 7.15). Hence, an example analysis use case starts selecting a cluster of single cell samples in the lower left half of SPLOM. This selection shows blue lines on the top half of SPLOM, indicating their weights on the first singular vector. Both of these studies are

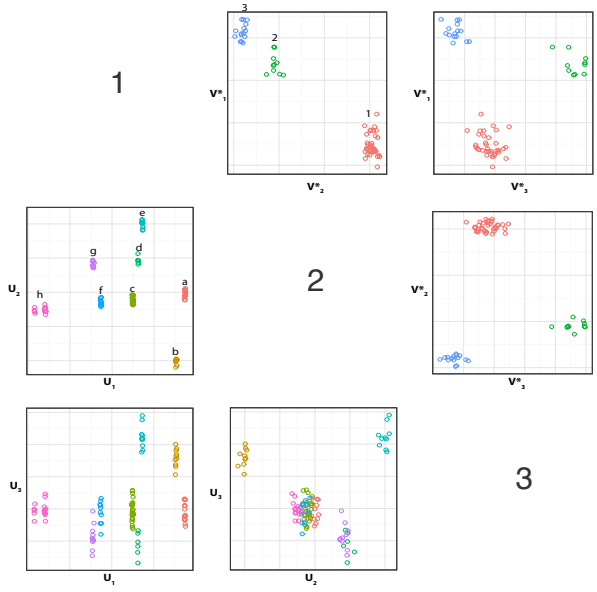

Figure 7.13: Scatter plot matrix of first three singular vectors. Known gene sets and sample cohorts are colour-coded (r= 3).

preliminary at large and requires further investigation to validate this biplot matrix vis idiom.

Another visualization design idea was to incorporate the singular values of the matrix Σ to scale the width and height for each scatterplot (Figure 7.16). In Figure 7.16A, the first eight singular vectors are compared, and each scatterplot has the same display dimension. On the other hand, the display dimensions for each singular vector are scaled based on its corresponding single value in Figure 7.16B. The preliminary feedback from the analysts was positive, and they thought the scaled scatterplots were more intuitive and avoided the bias introduced by coercing the scatter plot to a square dimension of the same size.

This design study of biplot matrix extended the existing vis idioms to exploit the outputs of SVD. Through a simple user interaction of selecting a set of samples or features, it allowed to explore the raw principal component scores and the variable component loadings. As also pointed out by [134], a weighted sum of original features (*loadings*) is a property that is often not fully exploited in PCA in the biological domain. The biplot matrix is a potential vis solution to exploit this property. The preliminary results with analysts are encouraging. Although the scatterplot is a fairly standard and common vis idiom, the biplot

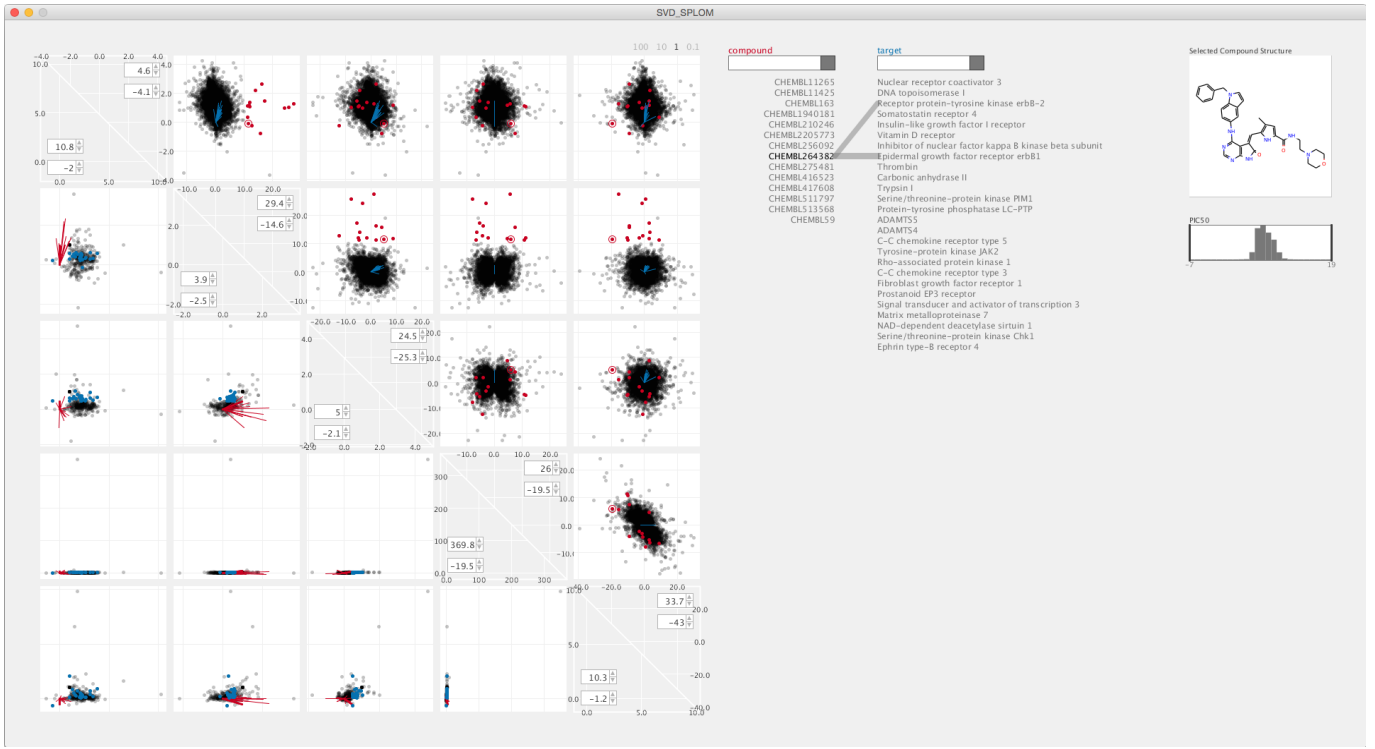

Figure 7.14: The interactive vis prototype for the drug target and compound dataset.

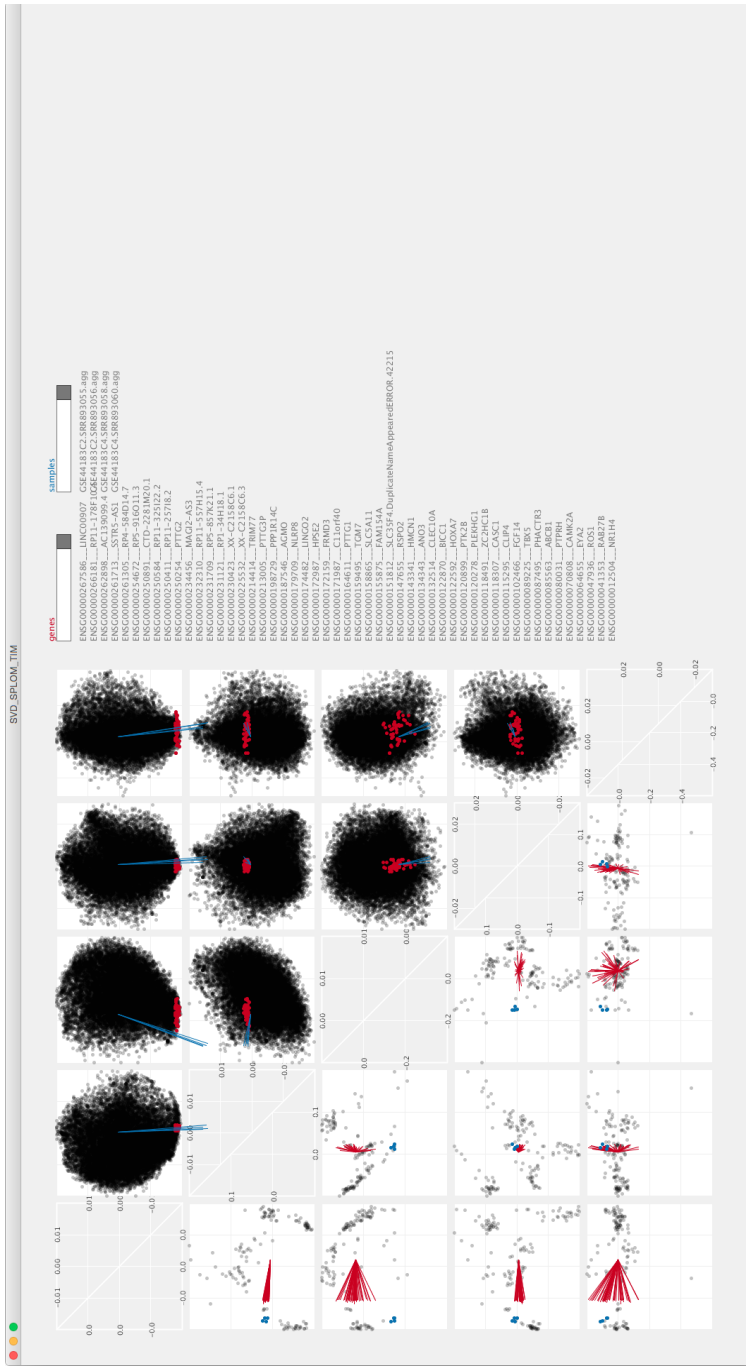

Figure 7.15: The interactive vis prototype for the gene expressions of single cell samples.

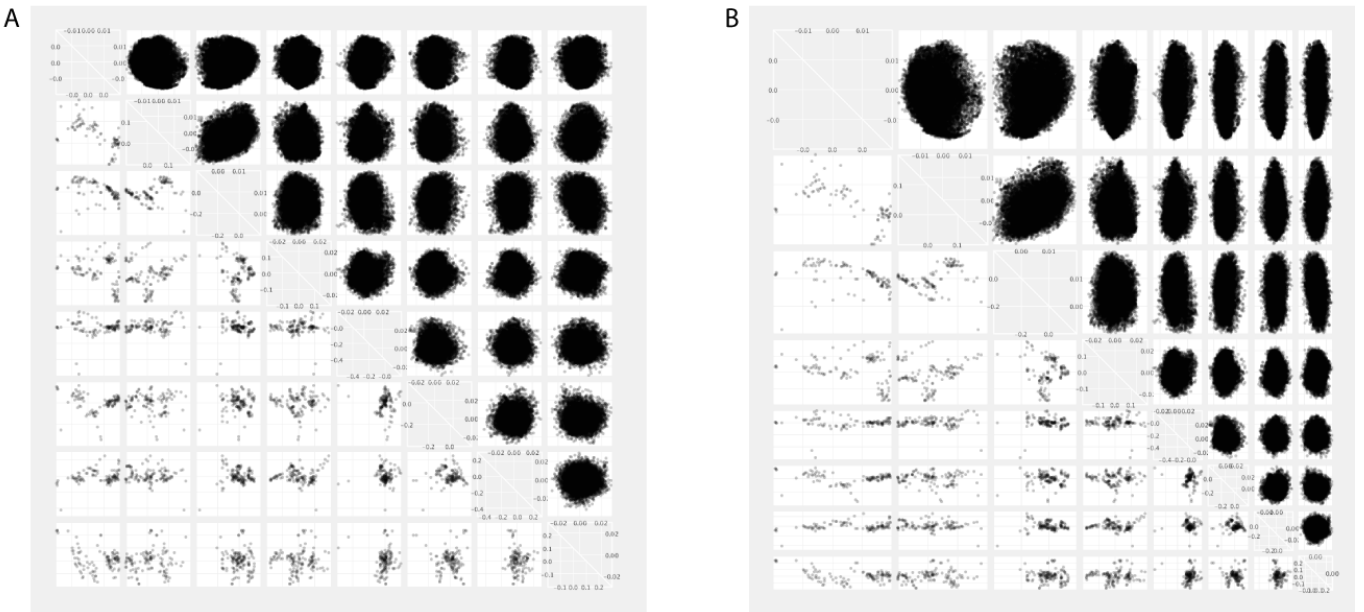

Figure 7.16: Comparison of biplot matrix vis idiom with and without scaling the display dimension for each scatterplot.

matrix vis idiom requires some background knowledge of the SVD method to utilise its interaction fully to explore the relationship between the sample and the feature spaces. The future work includes addressing the difficulty of interpreting the relation between the sample and the feature spaces. Also, we should consider incorporating more domain specific information, such as GO term enrichment or pathway analysis extensions for the gene expression data.

Chapter 8

Beyond Desktop Applications

8.1 Introduction

Although a display monitor of a personal computer is an obvious choice for developing data visualization systems for many analysis tasks, there are other mediums besides a screen to consider for communication and exploration of data. For example, there is a field of research in the physical representation of data, called *data physicalization*. In fact, data physicalization has been around for many years [135]. Especially with recent advances in digital fabrication, such as 3D printing technologies, it has become more accessible to produce tangible objects to represent physical or abstract data.

This chapter describes two projects that involved visualization mediums outside of desktop monitors. The first project is the Fly plot. Although the design process of Fly plot has already been discussed in Chapter 3, the use of printed visualization raises an interesting discussion about the interaction with the domain experts. The second project is a collaboration with Koen Van Mechelen, a Belgian artist, on the Cosmopolitan Chicken Research Project (CCRP). In this project, the single nucleotide polymorphism (SNP) array data of purebred and hybrid chickens were analysed and translated into data-driven sculptures. The 3D models were printed using selective laser sintering. The data sculptures were incorporated into the artist's exhibitions.

8.2 Case Study: Fly Plot in Print

Collaboration:

Bang Wong[1] and the Connectivity Map (CMap) research group[1]

B.W. conceived the study and jointly designed the visualization output. B.W. conducted the crowdsource pattern detection exercise.

[1]*Broad Institute of MIT and Harvard, Cambridge, MA, USA*

As discussed in Chapter 3, the data for Fly plot involved the measurement of gene expression for 1000 genes in response to 350 drug compounds at six different dosages measured in 12 different cell lines. Even with a small multiple for each gene and drug combination, there are 350,000 possible combinations. To scale down the number of possible permutations, the expert selected about 200 genes and 20 drugs. Then, all the 4000 combinations were plotted on an A0 poster size paper, and we invited domain experts to annotate the patterns they found interesting on the poster. This discussion was a very insightful and engaging exercise to learn about what kind of patterns they expected or did not expect, and why. This kind of discussion involving multiple domain experts and a visualization researcher would not have been practical and productive if the same poster were browsed on a computer screen.

Another interesting use of paper as a medium for data visualization was a crowdsource pattern detection exercise. During the presentation, our collaborator presented the project and explained the visual encoding of Fly plot. Then, he asked the participants to annotate any interesting patterns that the audience identified. This occasion was one of monthly, institution-wise meetings, and the room was full of scientists from a wide variety of biological domains. For this exercise, each A3 paper contained about 200 small multiples for 200 gene modulations based on a single drug compound. We preapred these sheets covering 150 drugs. The audience was asked to annotate any patterns they found interesting and give the reason. An example of annotated sheet is shown in Figure 8.1.

Both a poster and A3 crowdsourcing examples highlight the interesting potential of the paper medium to engage the users in pattern discovery. It is relatively inexpensive to generate these large figures with many small multiples, and these figures would have been less effective on the computer screen because of the limitations of the display and its resolution. We found the paper medium more engaging and effective in the discussion in a focus group setting. Even though our ultimate goal was to design a visualization tool on a computer display,

these intermediate steps with printed material provided invaluable insights into pattern detections and evaluation of visual encodings.

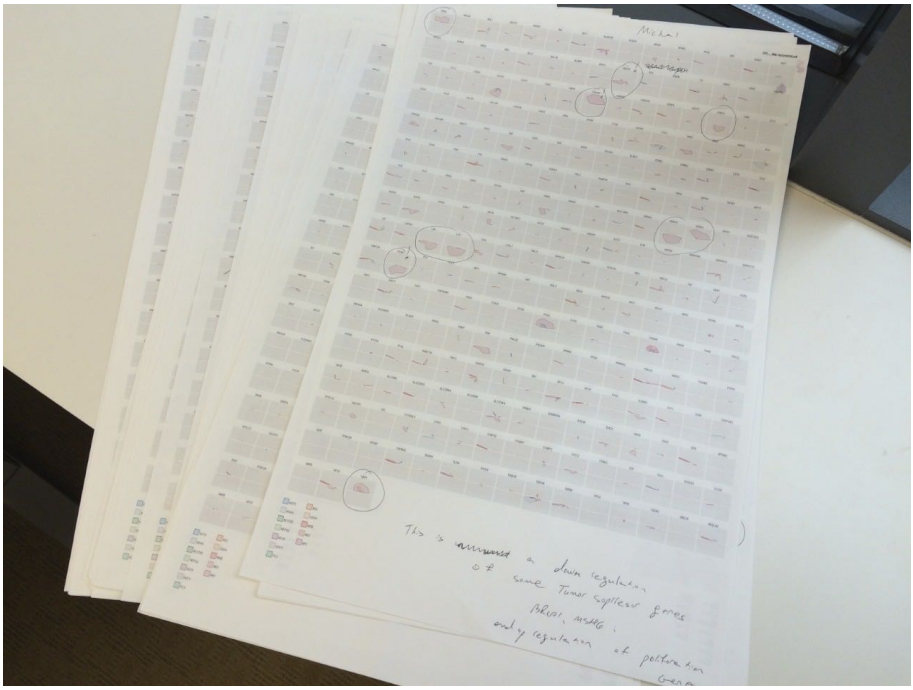

Figure 8.1: A photo of annotated crowdsourcing exercise sheets. Photo by Bang Wong.

8.3 Case Study: CCRP

Collaboration:

Koen Vanmechelen

The Cosmopolitan Chicken Research Project (CCRP) is a scientific research project [136], stemmed from the Cosmopolitan Chicken Project (CCP) [137]. Koen Vanmechelen, a Belgian conceptual artist, founded the CCP in the late nineties when he started crossbreeding domestic chickens from all over the world to reflect on the global culture and the genetic diversity. By April 2014, Vanmechelen had crossed 18 different inbred domestic chicken breeds with hybrid offspring in succession (Figure 8.2). The goal of CCRP is to explore both the heredity and genetic diversity from a scientific perspective, by analysing over 300 hybrids and inbred chickens to bridge art and science.

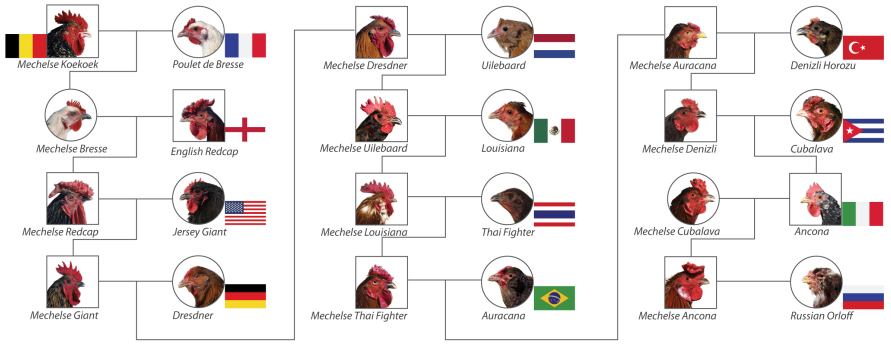

Figure 8.2: First twelve cross breeding of domesticated chickens. Images of chickens by Vanmechelen.

Genetic diversity arises at conception when two copies of each chromosome are brought together; one from the father and one from the mother. A chromosome is a structure of DNA and encodes most of genetic information. These chromosomes have small differences, called polymorphisms, which make each individual genetically unique. However, when crossing of related chickens in the process of domestication, many of these polymorphisms become fixed, meaning the chromosomes from the father and the mother become more similar. Mating of relatives is called inbreeding and leads to more uniform populations with fixed characteristics and less genetic diversity.

To analyse genetic diversity, we compare each position on a specific copy of a chromosome, referred to as an allele. If both alleles are the same, the chicken is said to be homozygous at that locus. If they are different, it is called heterozygous. Inbreeding leads to a higher degree of homozygosity and chicken breeders typically aim to make the alleles underlying desired traits homozygous. However, with high levels of inbreeding, some phenotypic disadvantages may occur. For example, an inbred chicken may have an impaired ability to adapt to a changing environment. Crossing different inbred animals leads to a higher degree of heterozygosity, and this has a positive effect on the survival rate and fertility of the crossbreeds. A nucleotide is a single DNA building block, and its variation is called a SNP. We detected SNPs for about 58,000 sites per sample using a SNP array experiment. Then, we processed the array data to determine genetic difference at each site (genotyping) and used the genotype information to construct data sculptures. The resulting data sculptures not only illustrated differences in the genetic profiles between inbred chickens, but also highlighted the degree of heterozygosity among crossbred chickens in comparison.

The chicken genome is about 40% the size of the human genome, and has a larger variability in chromosome size [138]. It consists of 39 pairs of chromosomes: 1 sex, 5 macro-, 5 intermediate, and 28 micro-chromosomes. Because the number of SNPs on chromosomes detected were scarce on chromosomes smaller than chromosome 28, only the genotype information from chromosome 1 through 28 were used for subsequent steps. In order to characterise the distribution of homozygous and heterozygous alleles along each chromosome, the occurrences of homozygous and heterozygous alleles were counted for each 100,000 base pair interval. The number of homozygous and heterozygous SNPs were translated into peaks in a circular layout (Figure 8.3). A peak pointing inwards represents the number of homozygous SNPs at a specific position on the chromosome, while an outward peak encodes the count of heterozygous SNPs. Figure 8.3 compares chromosomes one from two inbreds and a crossbred. Comparing *Mechelse Koekoek* and *Poulet de Bresse*, they both have not only a higher number of homozygous SNPs, but also distinct patterns of heterozygous regions. On the other hand, heterozygous regions are more widely distributed for the *Mechelse Ancona*. Each chromosome is represented as an arc of varying length to encode the chromosomal length. Each arc is connected at one end and rotated from this anchor point (Figure 8.4). The longest chromosomes, chromosomes 1 and 2, cross at the top of this spherical layout.

For processing the data and generating 3D models, we used Processing, an open source programming language and integrated development environment based on Java [70]. In addition, we used the toxiclibs library to make a watertight isosurface mesh of the structure and to export as STL for 3D printing [139]. Three STL files for *Mechelse Koekoek*, *Poulet de Bresse* and *Mechelse*

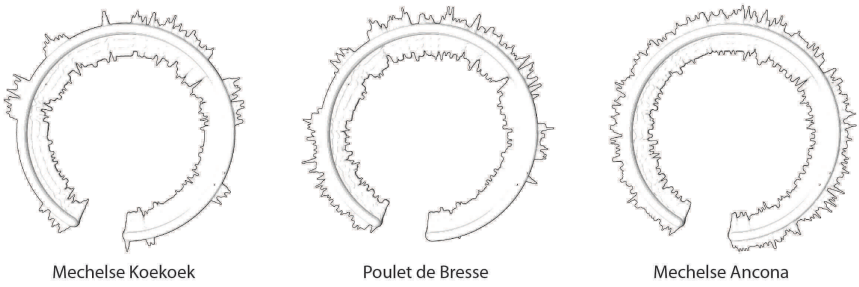

Figure 8.3: Visualising chromosome 1 of three chicken breeds. Peaks pointing outward represent the count of heterozygous SNPs, while peaks pointing inward represent the count of homozygous SNPs. *Mechelse Koekoek* and *Poulet de Bresse* are purebreds and *Mechelse Ancona* is a hybrid chicken.

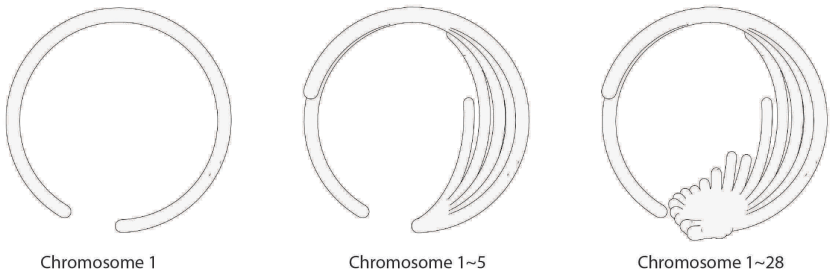

Figure 8.4: Illustration of chromosome arrangement for constructing a sculpture. Each tube-like arc represents a chromosome, and its length represents the size of the chromosome. Each chromosome starts at the same origin and is placed around to arrange 28 chicken chromosomes.

Ancona were generated (Figure 8.5). Data sculpture models were printed using the Selective Laser Sintering (SLS), an additive manufacturing technology where a high power laser fuses small powdered material to create a desired three-dimensional shape. The sculpture dimension is 23cm x 14cm x 28cm. Vanmechelen incorporated these data-driven sculptures into his exhibitions at Biennial of Venice and BOZAR Electronic Arts Festival in 2013 (Figure 8.6).

In the artist's view, the smooth representations of inbred lines correspond to weaker animals that are less adaptable to new environments (due to loss of heterozygosity), whereas the spiky sculptures correspond to animals that are more "aggressive" and ready to take on new challenges. Therefore, reaching out, diminishing the distance, longing for the other is dangerous and brave, but

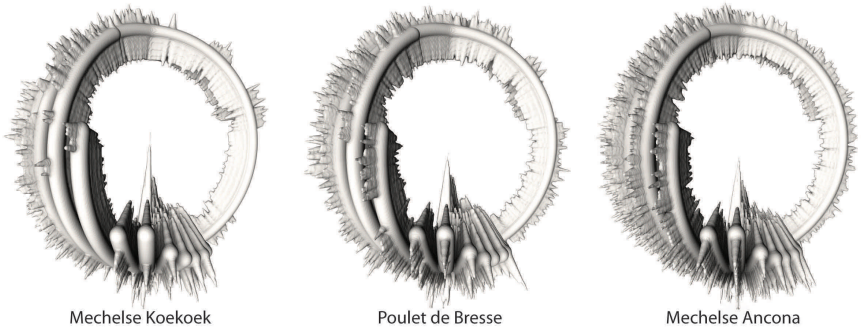

Figure 8.5: Renderings of 3D models for *Mechelse Koekoek*, *Poulet de Bresse* and *Mechelse Ancona*.

ultimately rewarding.

Although 3D printing technologies are becoming more accessible, it is still expensive to generate these 3D sculptures. The purpose of data sculpture is to engage the public audience to enjoy and to consume the scientific information about the genomic variations among inbred and crossbred chickens.

Figure 8.6: Photograph of data sculptures incorporated in Vanmechelen's art installation at Biennial of Venice in 2013. Image by Yu Chen, used with permission.

Chapter 9

Conclusion

9.1 Conclusion

Through a retrospective analysis of design studies from a wide range of biological application domains, it is evident that data visualization and visual analytics play a vital role in the data-intensive scientific paradigm. We presented eleven case studies and analysed the design process in retrospect to highlight different aspects of **the extended model of visualization** (Figure 2.5). Combined with the What-Why-How questions, this model provides a framework to generalise design rationales and decisions when discussing the design process of visualization systems. By carefully analysing each design study and calling attention to the process rather than the final product, we highlighted both scientific and engineering challenges in biological data visualization research.

As we rely more on computational and statistical methods to glean patterns and find relationships from large and complex datasets, visual analytics become more essential for data analysis and innovation of new analysis approaches. As seen in the dendsort case study (Chapter 6), interactive visualization prototypes, which were initially designed for data analysis, were instrumental for the subsequent development of leaf ordering methods. When users are provided with new ways to "see" and interact with the data, it also allows them to "think" differently about the data and the analysis approach. Thus, **well-designed visualization systems can augment our ability to exploit the sophisticated computational methods, to understand its performance, and possibly to innovate new approaches**.

With increasing complexity and heterogeneity of data, one key challenge

in science is to **draw connections between information from different sources to gain new insights**. For instance, Pipit (Chapter 3) and Oligoprobe (Chapter 7) integrated existing functional annotations, such as Gene Ontology (GO) terms and the Kegg pathway, to interpret the perceived patterns and relationships in the biological context. By presenting metadata in the context of the data under investigation, visualization systems support analytical reasoning tasks for the user. Thus, the challenge of biological data visualization extends beyond the detection of patterns to the interpretation of patterns by linking other information sources.

Conventional visualization techniques and practices should be challenged to re-evaluate and improve visual analysis and communication in science. For example, Weissgerber et al. call for a change in practice for presenting continuous data in small sample size studies [140]. The Point of View columns in Nature Methods [141, 142] also present practical design principles in the context of biology. Instead of employing existing visualization methods blindly, the user should consider the What-Why-How questions to examine the data, the task, and the vis idiom. As with any statistical methods, visualization methods need to be applied appropriately; otherwise, they lead to false insights or conclusions. To improve the current practice of data visualization, interaction between biological sciences and visualization research is essential to inform and advance each field.

A commonality between the vis design and the data-intensive biological science is that both embody **search problems**. In vis design, we explore possible visual encodings and interaction techniques to help the user carry out tasks more effectively [143]. On the other hand, the beginning of analysis in data-intensive science is inherently exploratory, searching for hunches and insights to develop into actionable hypotheses. Both are creative thinking processes that require exploring the edges of possibility that surround the domain problem (*adjacent possible*). As Johnson describes in his book [78], a good idea is not an attribute of a single breakthrough we colloquially describe these as "lightbulb moments", "eureka moments", or "epiphanies", but it is rather a network of conjectures and insights. **Visualization, as an external representation, serves as a cognitive tool to augment our perception, memory and information processing** [22]. Thus, visualization conveys relationships and patterns in data and serves as communications to others, as well as for oneself.

Another motivation for challenging conventional visualization techniques is to account for new analysis tasks which emerge from the accessibility of more data than ever before. Many of the conventional methods were developed before the advent of high-throughput experiments and modern high-speed computers. Sequence logo (Chapter 4), for example, has its strength in representing a motif; however it falls short when the task is to compare multiple alignment sets. As

more curated sequence alignment sets have become available, the users' tasks have evolved to include comparative analysis. **It is a vis designer's task to recognise the strengths and weaknesses of existing visualization methods against intended analysis task and to practice a principled approach to redesign vis systems**.

In this thesis, we reviewed design studies and analysed their design processes. Through this exercise, we extended the existing model of visualization and proposed the **4-step vis idiom design guideline** as a practical guideline for designing vis idioms for specialised domains. We also presented **data sketching** as a divergent design methodology (Chapter 4) and explained its key concepts including *early use of real data*, *early delivery of concepts*, and *iterative refinements*. We focused our discussion on *utility*, rather than *soundness* or *attractiveness* of the Vitruvius triangle [144]. Also, the functional effectiveness of design visualizations tools were not measured objectively. Instead, we included anecdotal evidence from the domain experts to validate the utility of our proposed design solutions. While qualitative evaluation approaches were more appropriate for custom visualization tools in our study, the quantitative approaches would be more suitable for evaluation of general purpose tools (Figure 2.7).

Designing an effective visualization system is difficult, and there are common pitfalls in problem-driven visualization research [6]. Learning and being vigilant of these pitfalls is useful, but to some extent, these lessons can only be learned from experience, and each project will always involve some risks. Ultimately, visualization research is a design challenge. "To design is to solve a problem. To be a designer is to be driven by the need to find something that needs solving and to do the work of coming up with the right solution" said Monteiro [145]. As Monteiro further elaborates, the best practice as a designer is to distrust all potential solutions until one reaches a high level of confidence, based on evidence.

9.2 Lessons Learned

The following is a list of advice for visualization researchers in a biological domain. Some may be obvious and considered common sense, but they are often easier said than done in practice. The views and opinions expressed here are based on the experience gained during this doctoral study.

9.2.1 Skills and Knowledge

There are three essential components for designing custom visualization tools: **rapid prototyping skills, knowledge of the application domain, and fundamental visualization principles**. As emphasised in this thesis, it is useful for a vis designer to acquire programming skills to implement functional prototypes quickly to engage the target end user in the design process. To design prototypes of custom visualization tools requires substantial knowledge in the application domain for data and task abstraction. How much domain knowledge is desired is debatable [6], but a designer should aim to develop the **domain specific intuition** [146] to address why a visualization system is useful for a specific context. For instance, attending regular meetings of the domain experts is a very useful way to learn about their research and how they communicate and present their findings. These meetings also help to build a rapport with collaborators. Needless to say, having an understanding of basic visualization principles [15] is useful to guide the initial design of vis idioms. Simon et al. introduced the role of "Liaison" for design study projects [147]. A "Liaison" has considerable expertise in both visualization and the application domain to "foster richer and more effective interdisciplinary communication in problem characterization, design and evaluation processes" [147]. Depending on group or project size, the "Liaison" may be an intermidiate role between biologists and engineers, a role that leans towards one side, or a role that encompassing both sides.

9.2.2 Design Study Guideline

Sedlmair et al. proposed a nine-stage methodological framework with potential pitfalls for each stage [6]. This framework covers the entire process of conducting a design study and provides practical guidance for vis designers to consider alternative approaches and ideas. In this thesis, we introduced the **4-step vis idiom design guideline**, which corresponds specifically to the *Design* stage in the Sedlmair's framework. The four steps consist of *Pop-out Effect*, *Effectiveness Principle*, *Pattern Expressiveness* and *Interactive Exploration*. Each step has different design considerations, characteristics, and goals, as discussed in Chapter 2. This guideline is not a formula, a recipe, or a linear process for an effective vis idiom. Rather, it should be considered as a set of considerations for examining design choices and fostering an introspective approach to vis idiom design. We hope that this vis idiom design guideline will entice further methodological discussion on different abstraction layers within the vis idiom design choices.

9.2.3 Visualization as a Process

We emphasise the development of a custom visualization tool as an iterative process in which the designer actively engages with the domain experts. Through iterations, the visualization tool is refined and tailored to the target users and their analysis needs. The level of unexpectedness in each iteration may be small, however, the iterative approach in search and evaluation of design choices allows exploration of the design space, as discussed with the concept of *adjacent possible* in the Sequence Diversity Diagram example (Chapter 4). It should be noted that this approach may not be suitable for the development of general purpose tools, as opposed to custom vis tools (Figure 2.7). Also, the tailored and novel custom visualization solution may require a certain level of visual learning and the specific domain knowledge from new users. Further longitudinal studies are required to investigate how much learning and domain knowledge is necessary for the adoption of novel visual encodings.

9.2.4 Environment and Work Culture

Because a collaboration with the end user is fundamental to **user-centered design** and design studies of custom visualization tools, it is useful to understand the work environment and culture of the collaborator. For example, the collaborator may be a Ph.D. student who works on his research project by himself or a group of researchers who work together on one project. We found the latter situation more accommodating for a new collaboration while the former may be more hesitant or reluctant to share their data even if a vis designer's intention is to help their analysis. Another important consideration is the physical distance between the vis designer and the domain experts. Even with a very promising beginning of a collaborative project, it is very difficult to maintain the momentum and the commitment to a project without being physically co-located. Thus, it is important for a vis designer to schedule regular meetings with the domain experts at the start of a collaborative project.

9.2.5 Design Contests

We advocate that visualization researchers participate in data visualization contests. There are a number of venues that host annual contests, such as BioVis and IEEE VAST challenges. Typically, these contests provide data analysis tasks in a specific context. Even if it is outside of one's research domain, it is a great opportunity to exercise your visualization design skills and to learn about other research domains. These present other advantages, such as

new opportunities for collaboration, the fixed deadline, networking with other researchers, opportunities to present the submission entry at conferences, and possible publications. Our contributions to the BioVis data contests have been recognised, leading to an invitation to organize the design contest for BioVis 2015. For these reasons, participating in contests can be very rewarding and beneficial for vis researchers.

9.2.6 Practice in the Wild

Visiting research institutes abroad has been one of the most valuable experiences as a vis designer. During the doctoral study, there were opportunities to visit the Institute for Systems Biology in Seattle for two months and the Broad Institute in Cambridge, Massachusetts for a one-week period twice. Practicing a role of vis designer in an entirely new environment gives a chance to test and train design skills. Also, working with a more experience designer as well as having them as a mentor is often the quickest way to learn design methods. Thus, we recommend vis researchers to seek actively for opportunities for research visits whenever you meet inspiring designers or encounter an interesting topic, at a conference for example.

9.2.7 Summary

To summarise, the items below are specific advice for future visualization researchers in a biological domain:

- Sketch on paper or digitally to explore the adjacent possible and the vis design space.
- Consider the three key concepts of data sketching: the early use of the real data, early delivery of concepts, and iterative refinement.
- Attend the regular meetings of your domain experts.
- Shorten the physical distance between your workspace and that of your domain experts.
- Participate in data visualization challenges.
- Seek opportunities for research visits and mentorships from experienced designers.

Bibliography

[1] J. Grey, "Jim Gray on eScience: A transformed scientific method," *The Fourth Paradigm*, pp. 17–31, 2009.

[2] J. W. Tukey, *Exploratory Data Aanalysis*. Pearson, 1977.

[3] E. R. Mardis, "The $1,000 genome, the $100,000 analysis?," *Genome medicine*, vol. 2, p. 84, jan 2010.

[4] C. Ware, *Visual Thinking: for Design*. Morgan Kaufmann, 2008.

[5] T. Munzner, "A nested model for visualization design and validation," in *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, pp. 921–928, 2009.

[6] M. Sedlmair, M. Meyer, and T. Munzner, "Design study methodology: Reflections from the trenches and the stacks," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 12, pp. 2431–2440, 2012.

[7] T. Munzner, *Visualization Analysis and Design*. CRC Press, 2014.

[8] K. Beck, M. Beedle, A. V. Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, S. Mellor, K. Schwaber, J. Sutherland, and D. Thomas, "Manifesto for Agile Software Development," 2001.

[9] J. van Wijk, "The Value of Visualization," in *VIS 05. IEEE Visualization, 2005.*, pp. 79–86, IEEE, 2005.

[10] S. Sinek, *Start with Why: How Great Leaders Inspire Everyone to Take Action*. Portfolio, 2011.

[11] S. McKenna, D. Mazur, J. Agutter, and M. Meyer, "Design activity framework for visualization design," *To Appear in IEEE TVCG (Proc. InfoVis)*, vol. 20, no. 12, pp. 2191–2200, 2014.

[12] C. Görg, L. Hunter, J. Kennedy, S. O. Donoghue, and J. J. V. Wijk, "Biological Data Visualization," *Dagstuhl Reports*, vol. 2, no. 9, pp. 131–164, 2012.

[13] M. Brehmer and T. Munzner, "A multi-level typology of abstract visualization tasks.," *IEEE transactions on visualization and computer graphics*, vol. 19, no. 12, pp. 2376–85, 2013.

[14] a. Treisman and S. Gormican, "Feature analysis in early vision: evidence from search asymmetries.," *Psychological review*, vol. 95, no. 1, pp. 15–48, 1988.

[15] C. Ware, *Information Visualization: Perception for Design*. San Francisco: Morgan Kaufmann Publishers Inc., 2004.

[16] J. Mackinlay, "Automating the design of graphical presentations of relational information," *ACM Transactions on Graphics*, vol. 5, no. 2, pp. 110–141, 1986.

[17] J. Bertin, *Semiology of Graphics*. The University of Wisconsin Press, 1983.

[18] J. Heer and M. Bostock, "Crowdsourcing Graphical Perception: Using Mechanical Turk to Assess Visualization Design," *Proceedings of the 28th Annual Chi Conference on Human Factors in Computing Systems*, pp. 203–212, 2010.

[19] S. S. Stevens, *Psychophysics: Introduction to its perceptual, neural, and social prospects*. 1975.

[20] W. S. Cleveland and R. McGill, "Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods," *Journal of the American Statistical Association*, vol. 79, no. 387, pp. pp. 531–554, 1984.

[21] A. Fisher, G. B. Anderson, R. Peng, and J. Leek, "A randomized trial in a massive online open course shows people don't know what a statistically significant relationship looks like, but they can learn," *PeerJ*, vol. 2, p. e589, 2014.

[22] B. Tversky, "What do Sketches say about Thinking," *Proceedings of AAAI spring symposium on sketch understanding*, pp. 205–210, 2002.

[23] M. Krzywinski and E. Savig, "Points of view: Multidimensional data," *Nature Methods*, vol. 10, no. 7, pp. 595–595, 2013.

[24] B. Bodenmiller, E. R. Zunder, R. Finck, T. J. Chen, E. S. Savig, R. V. Bruggner, E. F. Simonds, S. C. Bendall, K. Sachs, P. O. Krutzik, and G. P. Nolan, "Multiplexed mass cytometry profiling of cellular states perturbed by small-molecule regulators," *Nature Biotechnology*, vol. 30, no. 9, pp. 858–867, 2012.

[25] B. Shneiderman, "The eyes have it: a task by data type taxonomy for information visualizations," *Proceedings 1996 IEEE Symposium on Visual Languages*, pp. 336–343, 1996.

[26] D. a. Keim, "Information visualization and visual data mining," *IEEE Transactions on Visualization and Computer Graphics*, vol. 8, no. 1, pp. 1–8, 2002.

[27] L. Wilkinson, "The grammar of graphics," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 6, pp. 673–677, 2010.

[28] J. Heer, B. Shneiderman, and C. Park, "A taxonomy of tools that support the fluent and flexible use of visualizations," *Interactive Dynamics for Visual Analysis*, vol. 10, pp. 1–26, 2012.

[29] J. S. Yi, Y. A. Kang, J. Stasko, and J. Jacko, "Toward a deeper understanding of the role of interaction in information visualization.," *IEEE transactions on visualization and computer graphics*, vol. 13, no. 6, pp. 1224–31, 2007.

[30] M. Brehmer, M. Sedlmair, S. Ingram, and T. Munzner, "Visualizing dimensionally-reduced data: Interviews with analysts and a characterization of task sequences," *Submitted to ACM BELIV Workshop*, pp. 1–8, 2014.

[31] J. Heer, "Keynote at OpenVis Conference," 2015.

[32] B. Fry and C. Reas, "Processing [https://processing.org]," 2012.

[33] Processing Foundation, "p5.js [http://p5js.org]," 2015.

[34] S. Murray, *Interactive Data Visualization for the Web*. O'Reilly Media, 1st editio ed., 2013.

[35] E. Bertini, M. Stefaner, and M. Meyer, "Data Stories #54 Designing Exploratory Data Visualization Tools w/ Miriah Meyer," 2015.

[36] D. Lloyd and J. Dykes, "Human-centered approaches in geovisualization design: Investigating multiple methods through a long-term case study," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2498–2507, 2011.

[37] J. Nielsen and D. Sano, "SunWeb: user interface design for Sun Microsystem's internal Web," *Computer Networks and ISDN Systems*, vol. 28, pp. 179–188, 1995.

[38] D. Zimmerman and C. Akerelrea, "A group card sorting methodology for developing informational Web sites," *Proceedings. IEEE International Professional Communication Conference*, 2002.

[39] J. R. Wood and L. E. Wood, "Card Sorting: Current practices and beyond," *Journal of Usability Studies*, vol. 4, no. 1, pp. 1–6, 2008.

[40] N. Maiden, "Card sorts to acquire requirements," *IEEE Software*, vol. 26, no. June, pp. 85–86, 2009.

[41] L. Upchurch, G. Rugg, and B. Kitchenham, "Using card sorts to elicit web page quality attributes," *IEEE Software*, vol. 18, no. August, pp. 84–89, 2001.

[42] G. Rugg and P. McGeorge, "The sorting techniques: A tutorial paper on card sorts, picture sorts and item sorts," *Expert Systems*, vol. 22, no. 3, pp. 94–107, 2005.

[43] M. Meyer, M. Sedlmair, P. S. Quinan, and T. Munzner, "The nested blocks and guidelines model," *Information Visualization*, vol. 0, no. 0, pp. 1–16, 2013.

[44] S. Gerrard and J. Dickinson, "Women's working wardrobes: A study using card sorts," *Expert Systems*, vol. 22, no. 3, pp. 108–114, 2005.

[45] G. Martine and G. Rugg, "That site looks 88.46% familiar: Quantifying similarity of Web page design," *Expert Systems*, vol. 22, no. 3, pp. 115–120, 2005.

[46] K. Deibel, R. Anderson, and R. Anderson, "Using edit distance to analyze card sorts," *Expert Systems*, vol. 22, no. 3, pp. 129–138, 2005.

[47] T. Fossum and S. Haller, "Measuring card sort orthogonality," *Expert Systems*, vol. 22, no. 3, pp. 139–146, 2005.

[48] G. Kelly, *The psychology of personal constructs*. WW Norton, 1955.

[49] S. Fincher and J. Tenenberg, "Making sense of card sorting data," *Expert Systems*, vol. 22, no. 3, pp. 89–93, 2005.

[50] K. Beyer, Hugh and Holtzblatt, *Contextual design: defining customer-centered systems*. Elsevier, 1997.

[51] P. Medvedev, M. Stanciu, and M. Brudno, "Computational methods for discovering structural variation with next-generation sequencing.," *Nat. Methods*, vol. 6, no. 11 Suppl, pp. S13–20, 2009.

[52] P. J. Stephens, C. D. Greenman, B. Fu, F. Yang, G. R. Bignell, L. J. Mudie, E. D. Pleasance, K. W. Lau, D. Beare, and L. a. Stebbings, "Massive Genomic Rearrangement Acquired in a Single Catastrophic Event during Cancer Development," *Cell*, vol. 144, pp. 27–40, jan 2011.

[53] M. Meyer, T. Munzner, and H. Pfister, "MizBee: A multiscale synteny browsers," in *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, pp. 897–904, 2009.

[54] R. Sakai, A. Sifrim, A. Vande Moere, and J. Aerts, "TrioVis: a visualization approach for filtering genomic variants of parent-child trios.," *Bioinformatics (Oxford, England)*, pp. 1–2, jun 2013.

[55] S. Goodwin, J. Dykes, S. Jones, I. Dillingham, G. Dove, A. Duffy, A. Kachkaev, A. Slingsby, and J. Wood, "Creative user-centered visualization design for energy analysts and modelers," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2516–2525, 2013.

[56] F. Anscombe, "Graphs in Statistical Analysis," *The American Statistician*, vol. 27, no. 1, pp. 17–21, 1973.

[57] D. A. Norman, *The Design of Everyday Things*. Cambridge, Massachusetts: The MIT Press, revised an ed., 2013.

[58] M. Krzywinski, J. Schein, and Ä. Birol, "Circos: an information aesthetic for comparative genomics," *Genome Res.*, no. 604, pp. 1639–1645, 2009.

[59] N. Huang, I. Lee, E. M. Marcotte, and M. E. Hurles, "Characterising and predicting haploinsufficiency in the human genome.," *PLoS genetics*, vol. 6, p. e1001154, oct 2010.

[60] P. Stankiewicz and J. R. Lupski, "Structural variation in the human genome and its role in disease.," *Annu. Rev. Med.*, vol. 61, pp. 437–55, jan 2010.

[61] L. Feuk, A. R. Carson, and S. W. Scherer, "Structural variation in the human genome.," *Nat. Rev. Genet.*, vol. 7, pp. 85–97, mar 2006.

[62] J. O. Korbel, A. E. Urban, J. P. Affourtit, B. Godwin, F. Grubert, J. F. Simons, P. M. Kim, D. Palejev, N. J. Carriero, L. Du, B. E. Taillon, Z. Chen, A. Tanzer, a. C. E. Saunders, J. Chi, F. Yang, N. P. Carter,

M. E. Hurles, S. M. Weissman, T. T. Harkins, M. B. Gerstein, M. Egholm, and M. Snyder, "Paired-end mapping reveals extensive structural variation in the human genome.," *Science*, vol. 318, pp. 420–6, oct 2007.

[63] A. Herbig, G. Jäger, F. Battke, and K. Nieselt, "GenomeRing: alignment visualization based on SuperGenome coordinates.," *Bioinformatics*, vol. 28, pp. i7–15, jun 2012.

[64] C. Nielsen and B. Wong, "Points of view: Representing genomic structural variation," *Nat. Methods*, vol. 9, pp. 631–631, jun 2012.

[65] M. G. Reese, B. Moore, C. Batchelor, F. Salas, F. Cunningham, G. T. Marth, L. Stein, P. Flicek, M. Yandell, and K. Eilbeck, "A standard variation file format for human genome sequences.," *Genome Biol.*, vol. 11, p. R88, jan 2010.

[66] I. Lappalainen, J. Lopez, L. Skipper, T. Hefferon, J. D. Spalding, J. Garner, C. Chen, M. Maguire, M. Corbett, G. Zhou, J. Paschall, V. Ananiev, P. Flicek, and D. M. Church, "DbVar and DGVa: public archives for genomic structural variation.," *Nucleic acids research*, vol. 41, pp. D936–41, jan 2013.

[67] D. Karolchik, A. S. Hinrichs, T. S. Furey, K. M. Roskin, C. W. Sugnet, D. Haussler, and W. J. Kent, "The UCSC Table Browser data retrieval tool.," *Nucleic Acids Res.*, vol. 32, pp. D493–6, jan 2004.

[68] E. Birney, J. a. Stamatoyannopoulos, A. Dutta, R. Guigó, T. R. Gingeras, E. H. Margulies, Z. Weng, M. Snyder, E. T. Dermitzakis, R. E. Thurman, M. S. Kuehn, C. M. Taylor, S. Neph, C. M. Koch, S. Asthana, A. Malhotra, I. Adzhubei, J. a. Greenbaum, R. M. Andrews, P. Flicek, P. J. Boyle, H. Cao, N. P. Carter, G. K. Clelland, S. Davis, N. Day, P. Dhami, S. C. Dillon, M. O. Dorschner, H. Fiegler, P. G. Giresi, J. Goldy, M. Hawrylycz, A. Haydock, R. Humbert, K. D. James, B. E. Johnson, E. M. Johnson, T. T. Frum, E. R. Rosenzweig, N. Karnani, K. Lee, G. C. Lefebvre, P. a. Navas, F. Neri, S. C. J. Parker, P. J. Sabo, R. Sandstrom, A. Shafer, D. Vetrie, M. Weaver, S. Wilcox, M. Yu, F. S. Collins, J. Dekker, J. D. Lieb, T. D. Tullius, G. E. Crawford, S. Sunyaev, W. S. Noble, I. Dunham, F. Denoeud, A. Reymond, P. Kapranov, J. Rozowsky, D. Zheng, R. Castelo, A. Frankish, J. Harrow, S. Ghosh, A. Sandelin, I. L. Hofacker, R. Baertsch, D. Keefe, S. Dike, J. Cheng, H. a. Hirsch, E. a. Sekinger, J. Lagarde, J. F. Abril, A. Shahab, C. Flamm, C. Fried, J. Hackermüller, J. Hertel, M. Lindemeyer, K. Missal, A. Tanzer, S. Washietl, J. Korbel, O. Emanuelsson, J. S. Pedersen, N. Holroyd, R. Taylor, D. Swarbreck, N. Matthews, M. C. Dickson, D. J. Thomas, M. T. Weirauch, J. Gilbert, J. Drenkow, I. Bell, X. Zhao, K. G. Srinivasan, W.-K. Sung, H. S. Ooi, K. P. Chiu, S. Foissac,

T. Alioto, M. Brent, L. Pachter, M. L. Tress, A. Valencia, S. W. Choo, C. Y. Choo, C. Ucla, C. Manzano, C. Wyss, E. Cheung, T. G. Clark, J. B. Brown, M. Ganesh, S. Patel, H. Tammana, J. Chrast, C. N. Henrichsen, C. Kai, J. Kawai, U. Nagalakshmi, J. Wu, Z. Lian, J. Lian, P. Newburger, X. Zhang, P. Bickel, J. S. Mattick, P. Carninci, Y. Hayashizaki, S. Weissman, T. Hubbard, R. M. Myers, J. Rogers, P. F. Stadler, T. M. Lowe, C.-L. Wei, Y. Ruan, K. Struhl, M. Gerstein, S. E. Antonarakis, Y. Fu, E. D. Green, U. Karaöz, A. Siepel, J. Taylor, L. a. Liefer, K. a. Wetterstrand, P. J. Good, E. a. Feingold, M. S. Guyer, G. M. Cooper, G. Asimenos, C. N. Dewey, M. Hou, S. Nikolaev, J. I. Montoya-Burgos, A. Löytynoja, S. Whelan, F. Pardi, T. Massingham, H. Huang, N. R. Zhang, I. Holmes, J. C. Mullikin, A. Ureta-Vidal, B. Paten, M. Seringhaus, D. Church, K. Rosenbloom, W. J. Kent, E. a. Stone, S. Batzoglou, N. Goldman, R. C. Hardison, D. Haussler, W. Miller, A. Sidow, N. D. Trinklein, Z. D. Zhang, L. Barrera, R. Stuart, D. C. King, A. Ameur, S. Enroth, M. C. Bieda, J. Kim, A. a. Bhinge, N. Jiang, J. Liu, F. Yao, V. B. Vega, C. W. H. Lee, P. Ng, A. Yang, Z. Moqtaderi, Z. Zhu, X. Xu, S. Squazzo, M. J. Oberley, D. Inman, M. a. Singer, T. a. Richmond, K. J. Munn, A. Rada-Iglesias, O. Wallerman, J. Komorowski, J. C. Fowler, P. Couttet, A. W. Bruce, O. M. Dovey, P. D. Ellis, C. F. Langford, D. a. Nix, G. Euskirchen, S. Hartman, A. E. Urban, P. Kraus, S. Van Calcar, N. Heintzman, T. H. Kim, K. Wang, C. Qu, G. Hon, R. Luna, C. K. Glass, M. G. Rosenfeld, S. F. Aldred, S. J. Cooper, A. Halees, J. M. Lin, H. P. Shulha, X. Zhang, M. Xu, J. N. S. Haidar, Y. Yu, V. R. Iyer, R. D. Green, C. Wadelius, P. J. Farnham, B. Ren, R. a. Harte, A. S. Hinrichs, H. Trumbower, H. Clawson, J. Hillman-Jackson, A. S. Zweig, K. Smith, A. Thakkapallayil, G. Barber, R. M. Kuhn, D. Karolchik, L. Armengol, C. P. Bird, P. I. W. de Bakker, A. D. Kern, N. Lopez-Bigas, J. D. Martin, B. E. Stranger, A. Woodroffe, E. Davydov, A. Dimas, E. Eyras, I. B. Hallgrímsdóttir, J. Huppert, M. C. Zody, G. R. Abecasis, X. Estivill, G. G. Bouffard, X. Guan, N. F. Hansen, J. R. Idol, V. V. B. Maduro, B. Maskeri, J. C. McDowell, M. Park, P. J. Thomas, A. C. Young, R. W. Blakesley, D. M. Muzny, E. Sodergren, D. a. Wheeler, K. C. Worley, H. Jiang, G. M. Weinstock, R. a. Gibbs, T. Graves, R. Fulton, E. R. Mardis, R. K. Wilson, M. Clamp, J. Cuff, S. Gnerre, D. B. Jaffe, J. L. Chang, K. Lindblad-Toh, E. S. Lander, M. Koriabine, M. Nefedov, K. Osoegawa, Y. Yoshinaga, B. Zhu, and P. J. de Jong, "Identification and analysis of functional elements in 1% of the human genome by the ENCODE pilot project.," *Nature*, vol. 447, pp. 799–816, jun 2007.

[69] P. Danecek, A. Auton, G. Abecasis, C. a. Albers, E. Banks, M. a. DePristo, R. E. Handsaker, G. Lunter, G. T. Marth, S. T. Sherry, G. McVean, and R. Durbin, "The variant call format and VCFtools.," *Bioinformatics*

(Oxford, England), vol. 27, pp. 2156–8, aug 2011.

[70] C. Reas, B. Fry, and J. Maeda, *Processing: A Programming Handbook for Visual Designers and Artists*. The MIT Press, 2007.

[71] G. Lupi and S. Posavec, "Dear Data [http://www.dear-data.com]," 2015.

[72] G. Lupi and S. Posavec, "Keynote: Dear Data [https://vimeo.com/133608605]," 2015.

[73] B. Fry, *Visualizing Data: Exploring and Explaining Data with the Processing Environment*. O'Reilly Media, 2008.

[74] T. Schneider and R. Stephens, "Sequence logos: a new way to display consensus sequences," *Nucleic acids research*, vol. 18, no. 20, pp. 6097–6100, 1990.

[75] C. Shannon, "A Mathematical Theory of Communication," *The Bell System Technical Journal*, vol. XXVII, no. 3, pp. 379–423, 1948.

[76] R. Kosara, F. Bendix, and H. Hauser, "Parallel sets: interactive exploration and visual analysis of categorical data.," *IEEE transactions on visualization and computer graphics*, vol. 12, no. 4, pp. 558–68, 2006.

[77] B. Tversky and M. Suwa, "Thinking with sketches," Oxford University Press, 2009.

[78] S. Johnson, *Where Good Ideas Come From*. Riverhead Books, 2011.

[79] A. I. Roca, A. C. Abajian, and D. J. Vigerust, "ProfileGrids solve the large alignment visualization problem: influenza hemagglutinin example," *F1000Research*, jan 2013.

[80] M. Schmidt, "The Sankey Diagram in Energy and Material Flow Management," *Journal of Industrial Ecology*, vol. 12, pp. 82–94, feb 2008.

[81] H. Hofmann and M. Vendettuoli, "Common angle plots as perception-true visualizations of categorical associations.," *IEEE transactions on visualization and computer graphics*, vol. 19, pp. 2297–305, dec 2013.

[82] A. M. Waterhouse, J. B. Procter, D. M. a. Martin, M. Clamp, and G. J. Barton, "Jalview Version 2–a multiple sequence alignment editor and analysis workbench.," *Bioinformatics (Oxford, England)*, vol. 25, pp. 1189–91, may 2009.

[83] S. D. Dunn, L. M. Wahl, and G. B. Gloor, "Mutual information without the influence of phylogeny or entropy dramatically improves residue contact prediction.," *Bioinformatics (Oxford, England)*, vol. 24, pp. 333–40, feb 2008.

[84] The Jmol Team, "Jmol: an open-source Java viewer for chemical structures in 3D.," 2007.

[85] J. Gómez, L. J. García, G. a. Salazar, J. Villaveces, S. Gore, A. García, M. J. Martín, G. Launay, R. Alcántara, N. Del-Toro, M. Dumousseau, S. Orchard, S. Velankar, H. Hermjakob, C. Zong, P. Ping, M. Corpas, and R. C. Jiménez, "BioJS: an open source JavaScript framework for biological data visualization.," *Bioinformatics (Oxford, England)*, vol. 29, pp. 1103–4, apr 2013.

[86] S. Purcell, B. Neale, K. Todd-Brown, L. Thomas, M. A. R. Ferreira, D. Bender, J. Maller, P. Sklar, P. I. W. de Bakker, M. J. Daly, and P. C. Sham, "PLINK: a tool set for whole-genome association and population-based linkage analyses.," *American journal of human genetics*, vol. 81, no. 3, pp. 559–75, 2007.

[87] C. North, R. May, R. Chang, B. Pike, A. Endert, G. A. Fink, and W. Dou, "Analytic Provenance: Process + Interaction + Insight," in *29th Annual CHI Conference on Human Factors in Computing Systems, CHI 2011*, pp. 33–36, 2011.

[88] K. Pougach, A. Voet, F. a. Kondrashov, K. Voordeckers, J. F. Christiaens, B. Baying, V. Benes, R. Sakai, J. Aerts, B. Zhu, P. Van Dijck, and K. J. Verstrepen, "Duplication of a promiscuous transcription factor drives the emergence of a new regulatory network," *Nature Communications*, vol. 5, p. 4868, 2014.

[89] N. Gehlenborg, S. I. O'Donoghue, N. S. Baliga, A. Goesmann, M. a. Hibbs, H. Kitano, O. Kohlbacher, H. Neuweger, R. Schneider, D. Tenenbaum, and A.-C. Gavin, "Visualization of omics data for systems biology.," *Nature methods*, vol. 7, pp. S56–68, mar 2010.

[90] N. J. Higham, "Computing the nearest correlation matrix—a problem from finance," *IMA Journal of Numerical Analysis*, vol. 22, no. 3, pp. 329–343, 2002.

[91] R. F. F. S. C. R Development Core Team, "R: A Language and Environment for Statistical Computing," 2008.

[92] A. Genz and F. Bretz, *Computation of Multivariate Normal and t Probabilities*. 2009.

[93] A. Genz, F. Bretz, T. Miwa, X. Mi, F. Leisch, F. Scheipl, and T. Hothorn, *mvtnorm: Multivariate Normal and t Distributions*.

[94] A. Inselberg, "The plane with parallel coordinates," *The Visual Computer*, vol. 1, no. 4, pp. 69–91, 1985.

[95] C. Vehlow, J. Heinrich, F. Battke, D. Weiskopf, and K. Nieselt, "iHAT: the interactive Hierarchical Aggregation Table," in *Proceedings of the 2011 IEEE Symp. Biological Data Visualization (BioVis)*, 2011.

[96] A. Inselberg, *Parallel coordinates: Visual multidimensional geometry and its applications*. 2009.

[97] M. Eisen and P. Spellman, "Cluster analysis and display of genome-wide expression patterns," *Proceedings of the National Academy of Sciences*, no. 22, pp. 12930–12933, 1998.

[98] Y. Yao, J. Sun, X. Huang, G. R. Bowman, G. Singh, M. Lesnick, L. J. Guibas, V. S. Pande, and G. Carlsson, "Topological methods for exploring low-density states in biomolecular folding pathways.," *The Journal of chemical physics*, vol. 130, no. 14, p. 144115, 2009.

[99] M. Nicolau, A. J. Levine, and G. Carlsson, "Topology based data analysis identifies a subgroup of breast cancers with a unique mutational profile and excellent survival.," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 108, no. 17, pp. 7265–70, 2011.

[100] "Iris User Manual [http://www.ayasdi.com/index.php/user-manual]."

[101] G. Singh, F. Mémoli, and G. Carlsson, "Topological Methods for the Analysis of High Dimensional Data Sets and 3D Object Recognition," *Methods*, pp. 91–100, 2007.

[102] Z. Liu and J. Heer, "The Effects of Interactive Latency on Exploratory Visual Analysis," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 20, no. 12, pp. 2122–2131, 2014.

[103] R. Machiraju, W. Ray, and J. Bohland, "BioVis 2014 Data Contest," 2014.

[104] S. M. Smith, C. F. Beckmann, J. Andersson, E. J. Auerbach, J. Bijsterbosch, G. Douaud, E. Duff, D. a. Feinberg, L. Griffanti, M. P. Harms, M. Kelly, T. Laumann, K. L. Miller, S. Moeller, S. Petersen, J. Power, G. Salimi-Khorshidi, A. Z. Snyder, A. T. Vu, M. W. Woolrich, J. Xu, E. Yacoub, K. Uğurbil, D. C. Van Essen, and M. F. Glasser, "Resting-state fMRI in the Human Connectome Project.," *NeuroImage*, vol. 80, pp. 144–68, oct 2013.

[105] S. M. Smith, D. Vidaurre, C. F. Beckmann, M. F. Glasser, M. Jenkinson, K. L. Miller, T. E. Nichols, E. C. Robinson, G. Salimi-Khorshidi, M. W. Woolrich, D. M. Barch, K. Uğurbil, and D. C. Van Essen, "Functional connectomics from resting-state fMRI.," *Trends in cognitive sciences*, vol. 17, pp. 666–82, dec 2013.

[106] A. J. Bass, V. Thorsson, I. Shmulevich, S. M. Reynolds, M. Miller, B. Bernard, T. Hinoue, P. W. Laird, C. Curtis, H. Shen, D. J. Weisenberger, N. Schultz, R. Shen, N. Weinhold, D. P. Kelsen, R. Bowlby, A. Chu, K. Kasaian, A. J. Mungall, A. Gordon Robertson, P. Sipahimalani, A. Cherniack, G. Getz, Y. Liu, M. S. Noble, C. Pedamallu, C. Sougnez, A. Taylor-Weiner, R. Akbani, J.-S. Lee, W. Liu, G. B. Mills, D. Yang, W. Zhang, A. Pantazi, M. Parfenov, M. Gulley, M. Blanca Piazuelo, B. G. Schneider, J. Kim, A. Boussioutas, M. Sheth, J. a. Demchok, C. S. Rabkin, J. E. Willis, S. Ng, K. Garman, D. G. Beer, A. Pennathur, B. J. Raphael, H.-T. Wu, R. Odze, H. K. Kim, J. Bowen, K. M. Leraas, T. M. Lichtenberg, S. Weaver, M. McLellan, M. Wiznerowicz, R. Sakai, M. S. Lawrence, K. Cibulskis, L. Lichtenstein, S. Fisher, S. B. Gabriel, E. S. Lander, L. Ding, B. Niu, A. Ally, M. Balasundaram, I. Birol, D. Brooks, Y. S. N. Butterfield, R. Carlsen, J. Chu, E. Chuah, H.-J. E. Chun, A. Clarke, N. Dhalla, R. Guin, R. a. Holt, S. J. M. Jones, D. Lee, H. a. Li, E. Lim, Y. Ma, M. a. Marra, M. Mayo, R. a. Moore, K. L. Mungall, K. Ming Nip, J. E. Schein, A. Tam, N. Thiessen, R. Beroukhim, S. L. Carter, A. D. Cherniack, J. Cho, D. DiCara, S. Frazer, N. Gehlenborg, D. I. Heiman, J. Jung, J. Kim, P. Lin, M. Meyerson, A. I. Ojesina, C. Sekhar Pedamallu, G. Saksena, S. E. Schumacher, P. Stojanov, B. Tabak, D. Voet, M. Rosenberg, T. I. Zack, H. Zhang, L. Zou, A. Protopopov, N. Santoso, S. Lee, J. Zhang, H. S. Mahadeshwar, J. Tang, X. Ren, S. Seth, L. Yang, A. W. Xu, X. Song, R. Xi, C. a. Bristow, A. Hadjipanayis, J. Seidman, L. Chin, P. J. Park, R. Kucherlapati, S. Ling, A. Rao, J. N. Weinstein, S.-B. Kim, Y. Lu, G. Mills, M. S. Bootwalla, P. H. Lai, T. Triche Jr, D. J. Van Den Berg, S. B. Baylin, J. G. Herman, B. a. Murray, R. B. Arman Askoy, G. Ciriello, G. Dresdner, J. Gao, B. Gross, A. Jacobsen, W. Lee, R. Ramirez, C. Sander, Y. Senbabaoglu, R. Sinha, S. Onur Sumer, Y. Sun, V. Thorsson, L. Iype, R. W. Kramer, R. Kreisberg, H. Rovira, N. Tasman, D. Haussler, J. M. Stuart, R. G. W. Verhaak, M. D. M. Leiserson, B. S. Taylor, A. D. Black, J. Ann Carney, J. M. Gastier-Foster, C. Helsel, C. McAllister, N. C. Ramirez, T. R. Tabler, L. Wise, E. Zmuda, R. Penny, D. Crain, J. Gardner, K. Lau, E. Curely, D. Mallery, S. Morris, J. Paulauskis, T. Shelton, C. Shelton, M. Sherman, C. Benz, J.-H. Lee, K. Fedosenko, G. Manikhas, O. Potapova, O. Voronina, S. Belyaev, O. Dolzhansky, W. Kimryn Rathmell, J. Brzezinski, M. Ibbs, K. Korski, W. Kycler, R. Łaźniak, E. Leporowska, A. Mackiewicz, D. Murawa,

P. Murawa, A. Spychała, W. M. Suchorska, H. Tatka, M. Teresiak, R. Abdel-Misih, J. Bennett, J. Brown, M. Iacocca, B. Rabeno, S.-Y. Kwon, A. Kemkes, E. Curley, I. Alexopoulou, J. Engel, J. Bartlett, M. Albert, D.-Y. Park, R. Dhir, J. Luketich, R. Landreneau, Y. Y. Janjigian, E. Cho, M. Ladanyi, L. Tang, S. J. McCall, Y. S. Park, J.-H. Cheong, J. Ajani, M. Constanza Camargo, S. Alonso, B. Ayala, M. a. Jensen, T. Pihl, R. Raman, J. Walton, Y. Wan, G. Eley, K. R. Mills Shaw, R. Tarnuzzer, Z. Wang, L. Yang, J. Claude Zenklusen, T. Davidsen, C. M. Hutter, H. J. Sofia, R. Burton, S. Chudamani, and J. Liu, "Comprehensive molecular characterization of gastric adenocarcinoma.," *Nature*, 2014.

[107] N. Gehlenborg and B. Wong, "Points of view: Heat maps," *Nature Methods*, vol. 9, pp. 213–213, feb 2012.

[108] R. Suzuki and H. Shimodaira, "Pvclust: An R package for assessing the uncertainty in hierarchical clustering," *Bioinformatics*, vol. 22, no. 12, pp. 1540–1542, 2006.

[109] M. Meyer, B. Wong, M. Styczynski, T. Munzner, and H. Pfister, "Pathline: A tool for comparative functional genomics," *Computer Graphics Forum*, vol. 29, no. 3, pp. 1043–1052, 2010.

[110] C. Gilissen, A. Hoischen, H. G. Brunner, and J. a. Veltman, "Disease gene identification strategies for exome sequencing.," *Eur. J. Hum. Genet.*, vol. 20, pp. 490–7, may 2012.

[111] M. J. Bamshad, S. B. Ng, A. W. Bigham, H. K. Tabor, M. J. Emond, D. a. Nickerson, and J. Shendure, "Exome sequencing as a tool for Mendelian disease gene discovery.," *Nat. Rev. Genet.*, vol. 12, pp. 745–55, nov 2011.

[112] M. a. DePristo, E. Banks, R. Poplin, K. V. Garimella, J. R. Maguire, C. Hartl, A. a. Philippakis, G. del Angel, M. a. Rivas, M. Hanna, A. McKenna, T. J. Fennell, A. M. Kernytsky, A. Y. Sivachenko, K. Cibulskis, S. B. Gabriel, D. Altshuler, and M. J. Daly, "A framework for variation discovery and genotyping using next-generation DNA sequencing data.," *Nat. Genet.*, vol. 43, pp. 491–8, may 2011.

[113] The 1000 Genomes Project Consortium, "A map of human genome variation from population-scale sequencing," *Nature*, vol. 467, pp. 1061–1073, oct 2010.

[114] A. Sifrim, J. K. Van Houdt, L.-C. Tranchevent, B. Nowakowska, R. Sakai, G. a. Pavlopoulos, K. Devriendt, J. R. Vermeesch, Y. Moreau, and J. Aerts, "Annotate-it: a Swiss-knife approach to annotation, analysis and interpretation of single nucleotide variation in human disease.," *Genome Med.*, vol. 4, p. 73, sep 2012.

[115] K. Wang, M. Li, and H. Hakonarson, "ANNOVAR: functional annotation of genetic variants from high-throughput sequencing data.," *Nucleic Acids Res.*, vol. 38, p. e164, sep 2010.

[116] J. Goecks, A. Nekrutenko, and J. Taylor, "Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences.," *Genome Biol.*, vol. 11, p. R86, jan 2010.

[117] L. Wilkinson and M. Friendly, "The History of the Cluster Heat Map," *The American Statistician*, vol. 63, pp. 179–184, may 2009.

[118] M. C. P. de Souto, I. G. Costa, D. S. a. de Araujo, T. B. Ludermir, and A. Schliep, "Clustering cancer gene expression data: a comparative study.," *BMC bioinformatics*, vol. 9, p. 497, jan 2008.

[119] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. No. 2, 2009.

[120] P.-N. Tan, V. Kumar, and M. Steinbach, *Introduction to data mining*. Boston : Pearson Addison Wesley, 1st ed ed., 2005.

[121] Z. Bar-Joseph, D. K. Gifford, and T. S. Jaakkola, "Fast optimal leaf ordering for hierarchical clustering.," *Bioinformatics (Oxford, England)*, vol. 17 Suppl 1, pp. S22–9, jan 2001.

[122] S. a. Morris, B. Asnake, and G. G. Yen, "Dendrogram seriation using simulated annealing," *Information Visualization*, vol. 2, pp. 95–104, jun 2003.

[123] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning*, vol. 103 of *Springer Texts in Statistics*. New York, NY: Springer New York, 2013.

[124] G. Gruvaeus and H. Wainer, "Two Additions to Hierarchical Cluster Analysis," *Journal of Mathematical and Statistical Psychology*, vol. 25, pp. 200–206, 1972.

[125] M. Chae and J. J. Chen, "Reordering hierarchical tree based on bilateral symmetric distance.," *PloS one*, vol. 6, p. e22546, jan 2011.

[126] M. Hahsler, K. Hornik, and C. Buchta, "Getting Things in Order : An Introduction to the R Package seriation," *Journal Of Statistical Software*, vol. 25, no. 3, pp. 1–27, 2008.

[127] J. Quackenbush, "Computational analysis of microarray data.," *Nature reviews. Genetics*, vol. 2, pp. 418–27, jun 2001.

[128] C. Buchta, K. Hornik, and M. Hahsler, "Getting things in order: an introduction to the R package seriation," *Journal of Statistical Software*, vol. 25, no. 3, 2008.

[129] R Core Team and contributors worldwide, "R: Edgar Anderson's Iris Data [http://stat.ethz.ch/R-manual/R-patched/library/datasets/html/iris.html]."

[130] C. Ware, "Color sequences for univariate maps: theory, experiments and principles," *IEEE Computer Graphics and Applications*, vol. 8, pp. 41–49, sep 1988.

[131] E. R. Tufte, *The Visual Display of Quantitative Information*. Cheshire, CT, USA: Graphics Press, 1986.

[132] P. Klein, *Coding the Matrix: Linear Algebra through Applications to Computer Science*. Newtonian Press, 2013.

[133] K. R. Gabriel, "The Biplot Graphic Display of Matrices with Application to Principal Component Analysis," *Biometrika*, vol. 58, no. 3, pp. 453–467, 1971.

[134] R. Kolde and J. Vilo, "GOsummaries: an R Package for Visual Functional Annotation of Experimental Data," *F1000Research*, aug 2015.

[135] Y. Dragicevic, Pierre and Jansen, "List of Physical Visualizations [http://www.dataphys.org/list]."

[136] K. Vanmechelen, "Cosmopolitan Chicken Research Project [http://www.ccrp.be]."

[137] K. Vanmechelen, "Cosmopolitan Chicken Project [http://koenvanmechelen.be],"

[138] D. W. Burt, "Chicken genome: current status and future opportunities.," *Genome research*, vol. 15, pp. 1692–8, dec 2005.

[139] K. Schmidt, "toxiclibs [toxiclibs.org]."

[140] T. L. Weissgerber, N. M. Milic, S. J. Winham, and V. D. Garovic, "Beyond Bar and Line Graphs: Time for a New Data Presentation Paradigm," *PLOS Biology*, vol. 13, no. 4, p. e1002128, 2015.

[141] B. Wong, "Points of view: The design process," *Nature Methods*, vol. 8, no. 12, pp. 987–987, 2011.

[142] B. Wong, "Points of view: Design of data figures," *Nature Methods*, vol. 7, no. 9, pp. 665–665, 2010.

[143] M. Bostock, "Design is a Search Problem [https://youtu.be/fThhbt23SGM]," 2014.

[144] A. V. Moere and H. Purchase, "On the role of design in information visualization," *Information Visualization*, vol. 10, no. 4, pp. 356—-371, 2011.

[145] M. Monteiro, "Trust Nothing, Murder Most: Fighting for an idea is noble, until it's not.," 2015.

[146] C. W. Bartlett, S. Y. Cheong, L. Hou, J. Paquette, P. Y. Lum, G. Jäger, F. Battke, C. Vehlow, J. Heinrich, K. Nieselt, R. Sakai, J. Aerts, and W. C. Ray, "An eQTL biological data visualization challenge and approaches from the visualization community.," *BMC bioinformatics*, vol. 13 Suppl 8, p. S8, jan 2012.

[147] S. Simon, S. Mittelstädt, D. A. Keim, and M. Sedlmair, "Bridging the gap of domain and visualization experts with a Liaison," *Eurographics Conference on Visualization (EuroVis) - Short Papers*, 2015.

List of Publications

9.3 As First Author

R. Sakai and J. Aerts. Card Sorting Techniques for Domain Characterization in Problem-driven Visualization Research. *Eurographics Conf. Vis. - Short Pap.* 2015.

R. Sakai, R. Winand, T. Verbeiren, A. Vande Moere, and J. Aerts, "dendsort: modular leaf ordering methods for dendrogram representations in R," *F1000Research*, vol. 177, 2014.

R. Sakai and J. Aerts, "Erratum to : Sequence Diversity Diagram for comparative analysis of multiple sequence alignments," *BMC Proc.*, vol. 8, no. Suppl 2, p. S10, 2014.

R. Sakai and J. Aerts, "Sequence Diversity Diagram for comparative analysis of multiple sequence alignments," *BMC Proc.*, vol. 8, no. Suppl 2, p. S9, 2014.

R. Sakai, M. Moisse, J. Reumers, and J. Aerts, "Pipit: visualizing functional impacts of structural variations.," *Bioinformatics*, vol. 29, no. 17, pp. 2206–7, Sep. 2013.

R. Sakai, A. Sifrim, A. Vande Moere, and J. Aerts, "TrioVis: a visualization approach for filtering genomic variants of parent-child trios.," *Bioinformatics*, pp. 1–2, Jun. 2013.

9.4 As Co-author

Network TCGA Research, incl. **R. Sakai** , "Comprehensive Molecular Characterization of Gastric Adenocarcinoma," *Nature*, 2014.

K. Pougach, A. Voet, F. a. Kondrashov, K. Voordeckers, J. F. Christiaens, B. Baying, V. Benes, **R. Sakai** , J. Aerts, B. Zhu, P. Van Dijck, and K. J. Verstrepen, "Duplication of a promiscuous transcription factor drives the emergence of a new regulatory network," *Nat. Commun.* , vol. 5, p. 4868, 2014.

A. Sifrim, D. Popovic, L.-C. Tranchevent, A. Ardeshirdavani, **R. Sakai** , P. Konings, J. R. Vermeesch, J. Aerts, B. De Moor, and Y. Moreau, "eXtasy: variant prioritization by genomic data fusion," *Nat. Methods* , vol. 10, no. 11, pp. 1083–4, Nov. 2013.

G. a Pavlopoulos, P. Kumar, A. Sifrim, **R. Sakai** , M. L. Lin, T. Voet, Y. Moreau, and J. Aerts, "Meander: visually exploring the structural variome using space-filling curves," *Nucleic Acids Res.* , vol. 41, no. 11, Apr. 2013.

A. Sifrim, J. K. Van Houdt, L.-C. Tranchevent, B. Nowakowska, **R. Sakai** , G. a Pavlopoulos, K. Devriendt, J. R. Vermeesch, Y. Moreau, and J. Aerts, "Annotate-it: a Swiss-knife approach to annotation, analysis and interpretation of single nucleotide variation in human disease," *Genome Med.* , vol. 4, no. 9, p. 73, Sep. 2012.

C. W. Bartlett, S. Y. Cheong, L. Hou, J. Paquette, P. Y. Lum, G. Jäger, F. Battke, C. Vehlow, J. Heinrich, K. Nieselt, **R. Sakai** , J. Aerts, and W. C. Ray, "An eQTL biological data visualization challenge and approaches from the visualization community," *BMC Bioinformatics*, vol. 13 Suppl 8, no. Suppl 8, p. S8, Jan. 2012.

9.5 Awards

Award for Good Combination of Analysis and Visualization to Solve the Challenge, IEEE VAST Challenge 2015. Chicago, IL, USA.

Overall Favorite Data Contest Award at BioVis 2014, ISMB 2014, July 11-15, 2014. Boston, MA, USA.

Redesign contest honourable mention at BioVis 2013, for the submission "Visualizing Sequence Conservation in Protein Families". October 13-14, 2013.

Best Artwork Award, "Cosmopolitan Chicken Research Project", ISMB 2013

Second prize in the best defended/presented poster at Benelux Bioinformatics Conference(BBC) 2012, Nijmegen, the Netherlands.

Second prize in the best presentation for poster flash at Benelux Bioinformatics Conference(BBC) 2012, Nijmegen, the Netherlands.

Biologist's Favourite Award for data visualization contest at 1st IEEE Symposium on Biological data Visualization. 2011.